DOCUMENT TRACKING USING RFID (D'TRACK)

ROZAZLAINIDAWATI BINTI HAJI AHMAD

ELECTRICAL & ELECTRONICS ENGINEERING
UNIVERSITI TEKNOLOGI PETRONAS
JUNE 2010

DOCUMENT TRACKING
USING RADIO FREQUENCY IDENTIFICATION (RFID)
D'TraXX

By

ROZAZILAINIDAWATI BINTI HJ AHMAD

FINAL PROJECT REPORT

Submitted to the Electrical & Electronics Engineering Programme
in Partial Fulfilment of the Requirements
for the Degree
Bachelor of Engineering (Hons)
(Electrical & Electronics Engineering)

Universiti Teknologi Petronas
Bandar Seri Iskandar
31750 Tronoh
Perak Darul Ridzuan

CERTIFICATION OF APPROVAL

DOCUMENT TRACKING SYSTEM USING RFID (RFID)
D'TraXX

by

Rozazilainidawati Bt Hj Ahmad

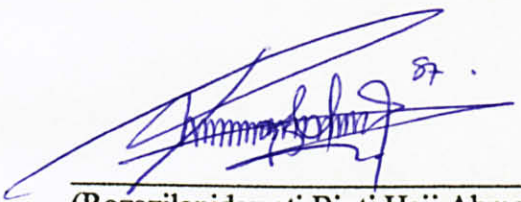
A project dissertation Submitted to the
Electrical & Electronics Engineering Programme
Universiti Teknologi PETRONAS
in partial fulfilment of the requirement for the
BACHELOR OF ENGINEERING (Hons)
(ELECTRICAL & ELECTRONICS ENGINEERING)

Approved by,

(Mrs. Hanita Daud)
Project Supervisor

CERTIFICATION OF ORIGINALITY

This is to certify that I am responsible for the work submitted in this project, that the original work is my own except as specified in the references and acknowledgements. Basically, I am continuing the project from the previous student. Therefore, certain content may include the past student's finding but it will be stated clearly in the reference.



(Rozazilanidawati Binti Haji Ahmad)

ABSTRACT

This project discusses the progress research done and basic understanding of the chosen topic, which is **Document Tracking System Using RFID (D'TraXX).** The purpose of this project is to improve the management of tracking documents using RFID as it is the right technology of choice to properly track the documents in order to save time and labour in finding documents and also proper maintenance of record of movement. This progress report will reflect problem statement, objectives and progress activity for the project. The equipments that are needed to achieve the objective of this project are RFID active tags, RFID reader, RFID antenna and document tracking software.

ACKNOWLEDGEMENTS

In the name of Allah, the Most Gracious, the Most Merciful, praise to Him the Almighty. Because of His will, I have managed to complete my Final Year Project. Therefore, I would like to take this opportunity to give my most gratitude to my supervisor, Puan Hanita Binti Daud for believe in me and giving me support, guidance, and always understand despite all the errors and delays that I have made during the completion of the project.

My appreciation also goes to all Final Year Project committee members and Electrical and Electronic Department staff for extensive knowledge and insightful advices; especially Dr. Yunus Bin Nayan. Without their help, it is hard for me to start and finish the work.

Besides that, I also would like to express my outstanding credit to the IT Programmer, Cik Norfadilah bt. Ahmad Mohari; UTP Master Student, Cik Siti Zarinah Bt Yusoff; Encik Shahjehan Bin Tahir; my colleagues, Ili Nadiah Binti Mohd Nasir, Syahira Nadia Binti Shaharuddin, Anith Safura Binti Azmi and Mohd Firdaus Bin Harris for their cooperation, knowledge sharing and ideas. I will embrace our friendship forever.

Last but not least, I furthermore would like to thank my family for their love and support while I was facing the hardship during completing the project. With full cooperation and encouragement from all above, I have successfully completed the project

Thank you.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

AC	ALTERNATING CURRENT
AIDC	AUTOMATIC IDENTIFICATION AND DATA CAPTURE
CCTV	CLOSED CIRCUIT TELEVISION
GUI	GRAPHICAL USER INTERFACE
ID	IDENTIFICATION
LOS	LINE OF SIGHT
MY SQL	MY STRUCTURED QUERY LANGUAGE
PC	PERSONAL COMPUTER
RF	RADIO FREQUENCY
RFID	RADIO FREQUENCY IDENTIFICATION
UTP	UNIVERSITI TEKNOLOGI PETRONAS
VB	VISUAL BASIC

CHAPTER 1
INTRODUCTION

1.1 Background of Study

Document tracking and management is a major area of market growth. RFID technology enables a break-through revolution in tracking documents. It is especially beneficial in those environments where the documents are of high value to the organization, and the temporary or permanent loss of a document would have significant negative impact. Examples are:

- Government offices
- Hospitals and other Medical offices
- Lawyer's offices
- Financial Institutions
- Educational Institutions

Not only do these industries maintain paper documentation regularly, but they have to keep such documentation for a considerable length of time.

1.2 Problem Statement

Finding an important document can be harder than finding a needle in a haystack. In Malaysia, financial institutions have to keep loan documents for up to 30 years plus, with the latter amount expected to increase in 10-12 years (RFID Applications: Document Tracking, 2006). Therefore, longer search time needed that leading to dissatisfaction.

Figure 1: Stack of documents

Besides that, there is no security validation for critical documents such as exam papers, financial document and etc. Moreover, there is also no record of time for those critical documents movement. They can be misfiled, or simply misplaced or stolen (Siti Ruzzana Binti Roslant, 2009). A method of knowing who had the last point of contact with the document would ensure the availability of documents at all times.

1.3 Objectives

My main objective through this project is to complete the system as an alternative for proper document management. The other objectives are as follows:

i. To improve the security of the current document management of UTP Strong Room by using RFID Tracking Document System (D'TraXX).

ii. To record the time and person who takes the critical documents in order to prevent misfiled, misplaced or stolen.

1.4 Scope of Study

RFID technology is simple, robust and inexpensive. The critical documents can be embedded with an RFID tag so the document can be tracked every time it is moved. Each RFID tag has a unique identification number and can be programmed with additional information such as type of media and storage location. If the document passes through an RFID door reader, the person will be prompted for identification, such as the user's name and ID, which is recorded. If an RFID-enabled document is reported missing, a complete history of all its users can be recalled along with specific dates and all the locations where the document has been taken (Ann Cavoukian, 2008).

The research and scope of study for this project is done in Universiti Teknologi PETRONAS (UTP). It is suitable to be used by the UTP Exam Unit as the document and data stored are very high value. Besides that, only qualified people can access to the document room. Therefore, we can increase the security. The basic idea for the RFID application to track document is simple:

- Each course has its own folder where the copies of the draft are kept until the final version; both soft and hard copies together with the answer scheme, course syllabus since all are subjected to audit at any time.
- Each document folder is tagged with an RFID tag.
- Staffs will be issued RFID cards as identification for checking documents in and out.
- Check In/Out Antennas and Readers are placed such a way so as to form an RFID grid within the premises or storage area.
- Graphical User Interface (GUI) is used to allow users to query document status, location, and history.
- User rights are given to access the documents.

CHAPTER 2
LITERATURE REVIEW

2.1 Introduction

The abbreviation *RFID* comes from Radio Frequency Identification, sometimes this technology is called an Automatic Identification and Data Capture *(AIDC)* and come into being as a natural upgrade of *Barcode Labels*, commonly used in 80's and of course nowadays (Łukasz Geldner & Paweł Nowiński, 2003).

Radio Frequency is a term that refers to alternating current (AC) having characteristics such that, if the current is input to an antenna, an electromagnetic field is generated suitable for wireless broadcasting and communications. These frequencies cover a significant portion of the electromagnetic radiation spectrum, extending from 9 kilohertz to thousands of gigahertz (Siti Ruzzana Binti Roslant, 2009).

Any radio frequency field has a wavelength that is inversely proportional to the frequency. In the atmosphere or in outer space, if f is the frequency in megahertz and s is the wavelength in meters, then

$$s = \frac{300}{f}$$

RFID is a contactless technology that uses radio frequency signals to transmit and receive data wirelessly, from a distance, from RFID tags or transponders to RFID readers (Ann Cavoukian, 2008). RFID is relatively new invention. First works, tests and ideas appeared in 1991, but first real solutions appeared in 1995 (Łukasz Geldner & Paweł Nowiński, 2003). RFID technology is generally used for automatic identification and to trigger processes that result in data collection or automation of manual processes (Ann Cavoukian, 2008). RF tags are available in a variety of shapes and sizes.

Figure 2 below shows an example of the RF tag.

Figure 2: RFID Tag

RFID is being employed to help manage and track document assets. In fact, about 35% of the document- tracking market in Malaysia is expected to use RFID by 2010. In the United States, several legal firms and tax courts, not to mention dental offices, have recently started tracking assets with RFID. Other potential uses would be in police departments, where paper-based case files are regularly maintained (RFID Applications: Document Tracking, 2006)

2.2 Frequency Ranges

The RFID system is distinguished by its frequency ranges. With the exception of the lowest-frequency segment, each band represents an increase of frequency corresponding to an order of magnitude (power of 10). Low frequency (30 KHz to 500 KHz) systems have short reading ranges and lower system costs. They are most commonly used in security access, asset tracking, and animal identification applications. High frequency (850 MHz to 950 MHz and 2.4 GHz to 2.5 GHz) systems, offering long read ranges (greater than 90 feet) and high reading speeds, are used for such applications as railroad car tracking an automated toll collection. However, the higher performance of high-frequency RFID systems incurs higher system costs (Siti Ruzzana Binti Roslant, 2009).

Table 1 depicts the eight bands in radio frequency spectrum, showing frequency and bandwidth ranges.

Table 1: Radio Frequency spectrum

Designation	Abbreviation	Frequencies	Free-space Wavelength
Very Low Frequency	VLF	9 kHz – 30 kHz	10 km – 100 km
Low Frequency	LF	30 kHz – 300 kHz	1 km – 10 km
Medium Frequency	MF	300 kHz – 3 MHz	100 m – 1 km
High Frequency	HF	3 MHz – 30 MHz	10 m – 100 m
Very High Frequency	VHF	30 MHz – 300 MHz	1 m – 10 m
Ultra High Frequency	UHF	300 MHz – 3 GHz	10 cm – 100 cm
Super High Frequency	SHF	3 GHz – 30 GHz	1 cm – 10 cm
Extremely Hugh Frequency	EHF	30 GHz – 300 GHz	1 mm – 10 mm

2.3 Basic component

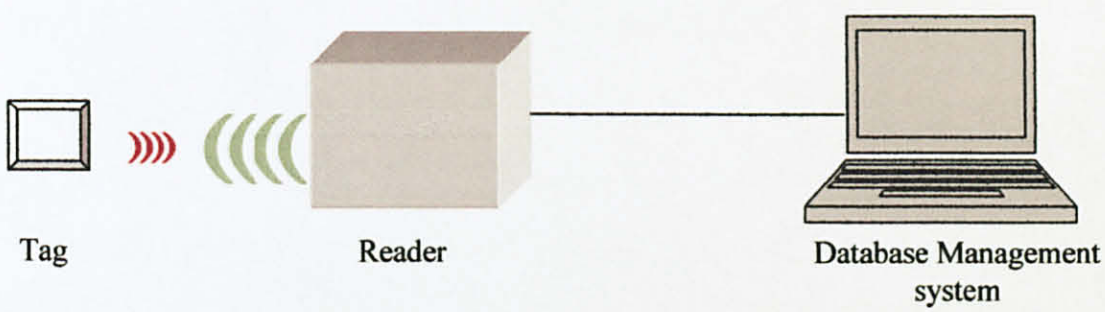


Figure 3: Basic components of RFID system.

A Radio-Frequency Identification system (RFID) has three component parts:

i. A transponder - the RFID tag - that has been programmed with unique information
ii. A reader or interrogator which is connected to an antenna that sends and receives the information
iii. A Database management system

2.3.1 RFID Tag/Transponder

Most RFID tags contain at least two parts. One is an integrated circuit for storing and processing information, modulating and demodulating a radio-frequency (RF) signal, and other specialized functions. The second is an antenna for receiving and transmitting the signal ("Radio-Frequency Identification", n.d.).

RFID tags have a wide variety of shapes and sizes. Animal tracking tags, inserted beneath the skin, can be as small as a pencil lead in diameter and one-half inch in length. Tags can be screw-shaped to identify trees or wooden items, or credit-card shaped for use in access applications ("RFID / What is RFID / Component", n.d). Examples of RFID tags are:

Figure 4: Polymer Flexible RFID tag

Figure 5: RFID tag use in electronic toll collection

Besides that, RFID tags may be of one of two types: active or passive. Active RFID tags have their own power source whereas passive RFID tags do not have batteries, can be much smaller, and have a virtually unlimited life span.

2.3.2 RFID Reader/interrogator

RFID reader is an electronic device used for communication between RFID tags and a host computer system. A reader generally consists of an RF transmitter and receiver and an antenna for communicating with tags. A digital interface enables the reader to communicate with the host computer system.

Often the antenna is packaged with the transceiver and decoder to become a reader, which can be configured either as a handheld or a fixed-mount device. The reader emits radio waves in ranges of anywhere, depending upon its power output and the radio frequency used. When an RFID tag passes through the electromagnetic zone, it detects the reader's activation signal. The reader decodes the data encoded in the tag's integrated circuit and the data is passed to the host computer for processing (Siti Ruzzana Binti Roslant, 2009). Examples of RFID antenna are:

Figure 6: High Frequency Reader RFID Writer with Antenna

Figure 7: Portable Handheld RFID Reader

2.3.3 Operating Frequency

The radio frequencies at which a tag transmits and receives signals have implications for the ability of the tag's signal to penetrate materials. As a general rule, higher frequencies are less able to penetrate substances such as metals or liquids than lower frequencies. Depending on the application, the penetration capabilities of a particular frequency can be either a benefit or a shortcoming (Guidelines for Securing RFID Systems, 2007).

Table 2: RFID operating frequencies and associated characteristics.

Band	LF Low Frequency	HF High Frequency	UHF Ultra High Frequency	Microwave
Frequency	30 - 300 kHz	3 - 30 MHz	300 - 1 GHz	2 -30 GHz
Typical RFID Frequency	125 -134 kHz	13.56 MHz	433 MHz or 865 956MHz	2.45 GHz
Approximate Read Range	Less than 0.5m	Up to 1.5m	433 MHz = up to 100m 865 - 956 MHz = 0.5 to 5m	Up to 10m
Typical Data Transfer Rate	Less than 1 kilobit per second (kbit/s)	Approximate 25 kbits/s	30 kbit/s	Up to 100 kbit/s
Characteristics	Short-range, low data transfer rate, penetrates water but not metal.	Higher ranges, reasonable data transfer rate (similar to GSM phone), penetrates water but not metal.	Long ranges, high data transfer rate, concurrent read of less than 100 items, cannot penetrate water or metals	Long ranges, high data transfer rate, cannot penetrate water or metal
Typical Use	Animal ID, Car, Immobiliser	Smart labels, Contact-less, Travel Cards Access and Security	Specialist Animal Tracking	Moving vehicle toll

As a general rule, radio signals at lower frequencies will propagate farther than signals at higher frequencies, assuming similar transmitter power levels. The attenuation (or decrease) of a radio signal as it travels through a medium such as air is directly related to its wavelength. All signals experience the same decrease in signal strength *per wavelength* when traveling through the same medium. Due to signals at lower frequencies have longer *wavelengths*; the signal attenuation occurs at a slower rate.

The figure below shows some of the common and less-common frequency bands in which RFID systems operate. Also shown is the corresponding wavelength - the distance between points at which the field has a fixed value when the signal moves at the velocity of light (Daniel M. Dobkin, 2005).

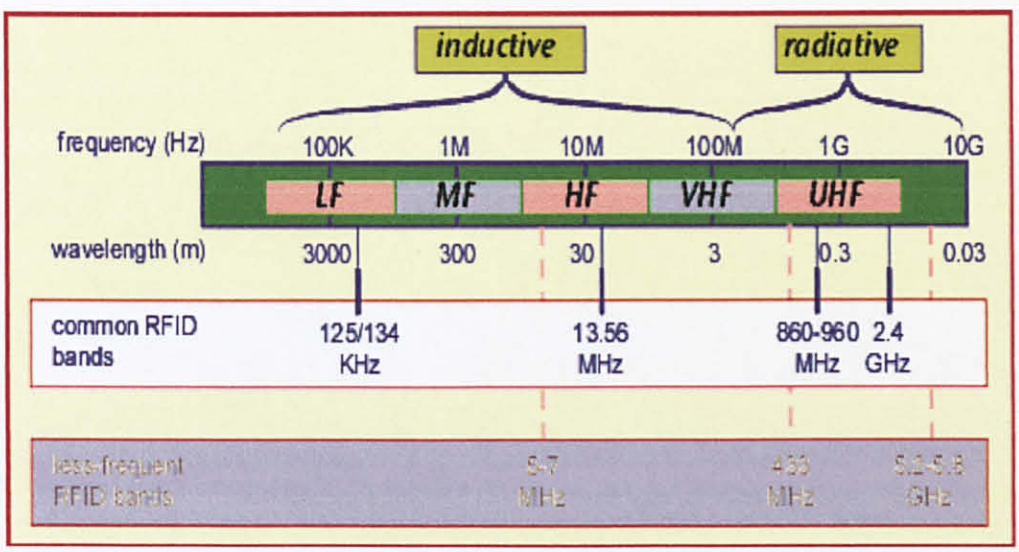


Figure 8: Common and less-common frequency bands in which RFID systems operate

Several issues are involved in choosing a frequency of operation. The most fundamental, as indicated in the Figure 8, is whether inductive or radiative coupling will be employed.

The distinction is closely related to the size of the antennas to be used relative to the wavelength. When the antennas are very small compared to the wavelength, the effects of the currents flowing in the antenna cancel when viewed from a great distance, so there is no radiation. Only objects so close to the antenna that one part of the antenna appears significantly closer than another part can feel the presence of the current. Thus, these systems, which are known as inductively-coupled systems, are limited to short ranges comparable to the size of the antenna (Daniel M. Dobkin, 2005). The systems at lower frequencies, such as 13.56 MHz, depend on inductive coupling as the primary mode of interaction. The range of an inductively coupled system drops sharply with distance, making communication beyond 10 to 20 feet impractical. Using longer-range electrical coupling at these frequencies is not recommended due to their high susceptibility to noise and interference from other devices.

Besides that, the radiative systems use antennas comparable in size to the wavelength. The very common 900 MHz range has wavelengths around 33 cm. Reader antennas vary in size from around 10 to 30 cm and tags are typically 10 to 18 cm long. These systems use radiative coupling, and are not limited by reader antenna size but by signal propagation issues (Daniel M. Dobkin, 2005).

The ability for signals to propagate within crowded environments is also dependent on the signal wavelength, and hence frequency. Within warehouses, truck yards, and other facilities, the ability for an RFID system to operate in and around obstructions is critical. These obstructions are often metal, such as vehicles, requiring signals to propagate "around" rather than "through" the obstructions. Active RFID signals propagate "around" obstructions by means of diffraction, and the level of diffraction is dependent on the size of the object versus the signal wavelength. Diffraction occurs when the wavelength approaches the size of the object.

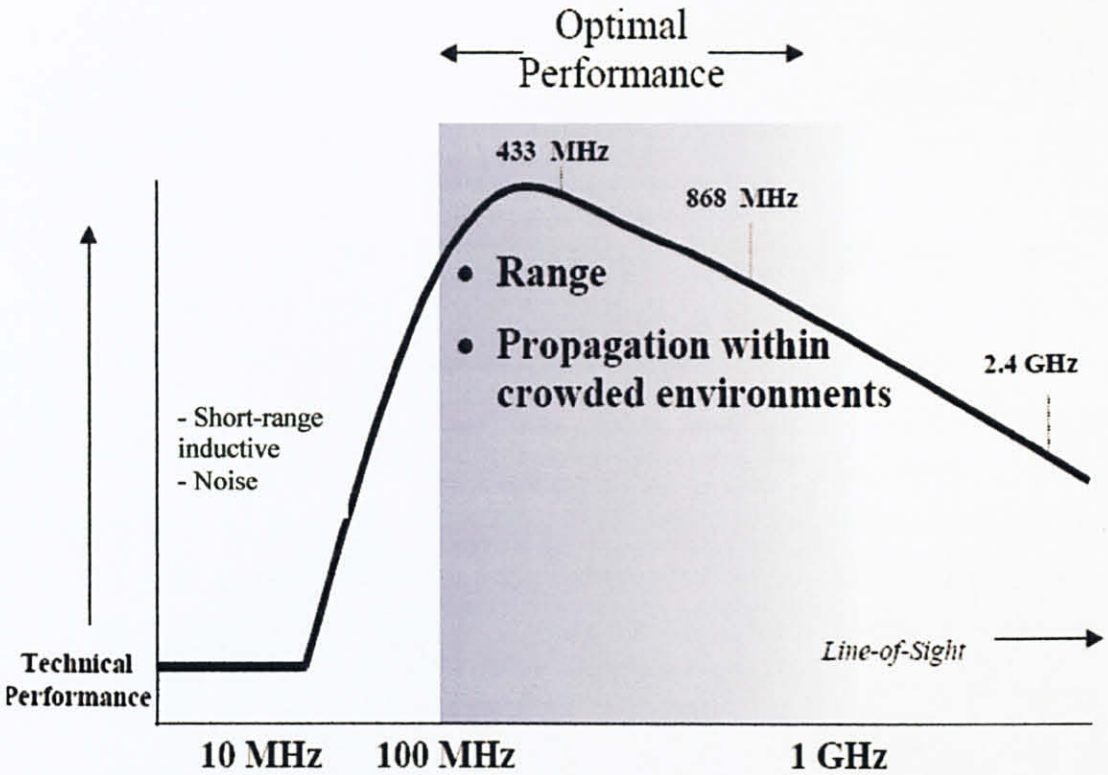


Figure 9: Frequencies between 100 MHz and 1 GHz offer the best technical performance for Active RFID.

For example, at 433 MHz the wavelength is approximately a meter, enabling signals to diffract around large obstructions. At 2.4 GHz, the wavelength is approximately a tenth of a meter and diffraction is very limited and creates blind spots and areas of limited coverage. Frequencies above 2 GHz present significant challenges for operation in crowded environments and are therefore not recommended for most RFID applications (Active and Passive RFID, 2009).

2.3.4 Working Principle

The three primary components that make RFID "Work" are Tags, Readers, and Software that collects data from the readers and helps turn it into actionable information (How RFID "Works", n.d).

Generally, for RFID to work, the antenna emits radio signals to activate the tag and to read and write data to it. Antennas are the conduits between the tag and the transceiver, which controls the system's data acquisition and communication. The electromagnetic field produced by an antenna can be constantly present when multiple tags are expected continually. If constant interrogation is not required, a sensor device can activate the field. Often the antenna is packaged with the transceiver and decoder to become a reader, which can be configured either as a handheld or a fixed-mount device. The reader emits radio waves in ranges of anywhere, depending upon its power output and the radio frequency used ("RFID / What is RFID / Component", n.d).

Basically, for a passive tag, when it passes through an electromagnetic field within reader, the tag is then powered on and transmits its information to the interrogator. The reader decodes the data encoded in the tag's integrated circuit. The information then is automatically sent into the Data System. Information exchange is done via air interface, precisely through electromagnetic waves, so without physical contact with reader (Łukasz Geldner & Paweł Nowiński, 2003).

If compared to the passive tag which is only powered on when passing through the reader's field, the active tag is always power on or continuously energized. This is due to the battery which it posses. Active tag has a read range of up to 300', but when the battery runs out, it will stop working.

2.4 RFID Types

2.4.1 Active RFID

Active RFID is a long range communication approach that has a reading distance between 50 m (150 feet) to 100 m (300 feet). Tags are powered by an internal battery and are typically read/write, where tag data can be rewritten and/or modified. An active tag's memory size varies according to application requirements; some systems operate with up to 1MB of memory.

In a typical read/write RFID work-in-process system, a tag might give a machine a set of instructions, and the machine would then report its performance to the tag. This encoded data would then become part of the tagged part's history. The battery-supplied power of an active tag generally gives it a longer read range. The trade off is greater size, greater cost, and a limited operational life (which may yield a maximum of 10 years, depending upon operating temperatures and battery type) ("RFID / What is RFID / Component", n.d).

2.4.2 Passive RFID

Tags operate without a separate external power source and obtain operating power generated from the reader. Passive tags are consequently much lighter than active tags, less expensive, and offer a virtually unlimited operational lifetime. The trade off is that they have shorter read ranges than active tags and require a higher-powered reader. Read-only tags are typically passive and are programmed with a unique set of data (usually 32 to 128 bits) that cannot be modified. Read-only tags most often operate as a license plate into a database, in the same way as linear barcodes reference a database containing modifiable product-specific information.

2.4.3 Differences between active and passive RFID

Active RFID and Passive RFID technologies, while often considered and evaluated together, are fundamentally distinct technologies with substantially different capabilities. In most cases, neither technology provides a complete solution (Active and Passive RFID, 2009). The majority of the RFID tags in use today are of the passive variety. Active RFID tag technology is still evolving and its widespread availability is expected to increase over the next several years ("Solving the New Technology Requirements for RFID Business Applications", 2009).

Table 3: Technical differences between Active and Passive RFID technologies.

	Passive RFID	**Active RFID**
Tag Power Source	External (Energy transferred from reader through RF)	Internal (Battery)
Tag Readability	Only within the area covered by the reader, typically up to 3 meters.	Can provide signals over an extended range, typically up to 100 meters.
Energization	A passive tag is energized only within field of reader.	An active tag is always energized.
Magnetic Field Strength	High, since the tag draws power from the electromagnetic field provided by the reader.	Low, since the tag emits signals using internal battery source.
Available Signal Strength from Tag to Reader	Low	High
Shelf Life	Very high, ideally does not expire over a life time.	Limited to about 5 years, the life of a battery.
Data storage	Limited data storage, typically 128 bytes.	Can store larger amounts of data.
Cost	Cheap	Expensive
Size	Smaller	Slightly bulky (due to battery)

Table 4: Summary of functional capabilities of Active and Passive RFID technologies.

	Active RFID	Passive RFID
Communication Range	Long range (100m or more)	Short or very short range (30m or less)
Multi-Tag Collection	Collects 1000s of tags over a 7 acre region from a single reader	Collects hundreds of tags within 3 meters from a single reader
	Collects 20 tags moving at more than 100mph	Collects 20 tags moving at 2mph[2] or slower
Sensor Capability	Ability to continuosly monitor and record sensor input; data/time stamp for sensor events	Ability to read and transfer sensor values only when tag is powered by reader; no date/time stamp.
Data Storage	Large read/write data storage (128kb) with sophisticated data search and access capabilities available	Small read/write data storage (e.g. 128 bytes)

2.5 Current Document Tracking Application

2.5.1 GAO RFID Asset Tracking's LocateWare

GAORFID Inc. announces, LocateWare(TM), an RFID middleware that is cost effective, easy to setup and configure and reports real-time location of people, animals and things. LocateWare(TM) enables comprehensive tracking, locating and identification for a wide range of businesses. LocateWare(TM) working in conjunction with RFID readers and RFID tags can significantly improve the utilization of critical resources and reduces search time and increases overall efficiency and security.

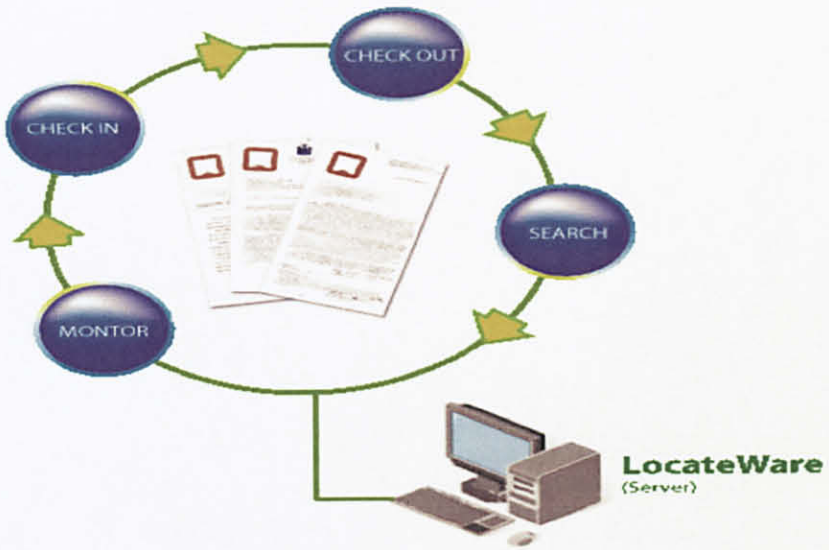


Figure 10: LocateWare system overview for document tracking

LocateWare, as shown in Figure 10, together with GAO RFID Asset Tracking's RFID readers successfully capture and distribute RFID tag information to the user in a customized report format for their unique job requirements. Document trays with RFID readers can hold as many as one hundred documents that can be read at one time. The number of documents read simultaneously can grow to the thousands just by adding more antennae in the read vicinity (Ken Cheung, 2007).

2.5.2 *DocuTrack3000*

DocuTrack3000 is a total RFID solution for document tracking as shown in Figure 11. It includes RFID labels, RFID reader and a document tracking software. DocuTrack3000 is designed to keep track of electronic documents and hard copy documents such as project file, document file and many others. Books and file can be tracked in the application in details ("Solving the New Technology Requirements for RFID Business Applications", 2009).

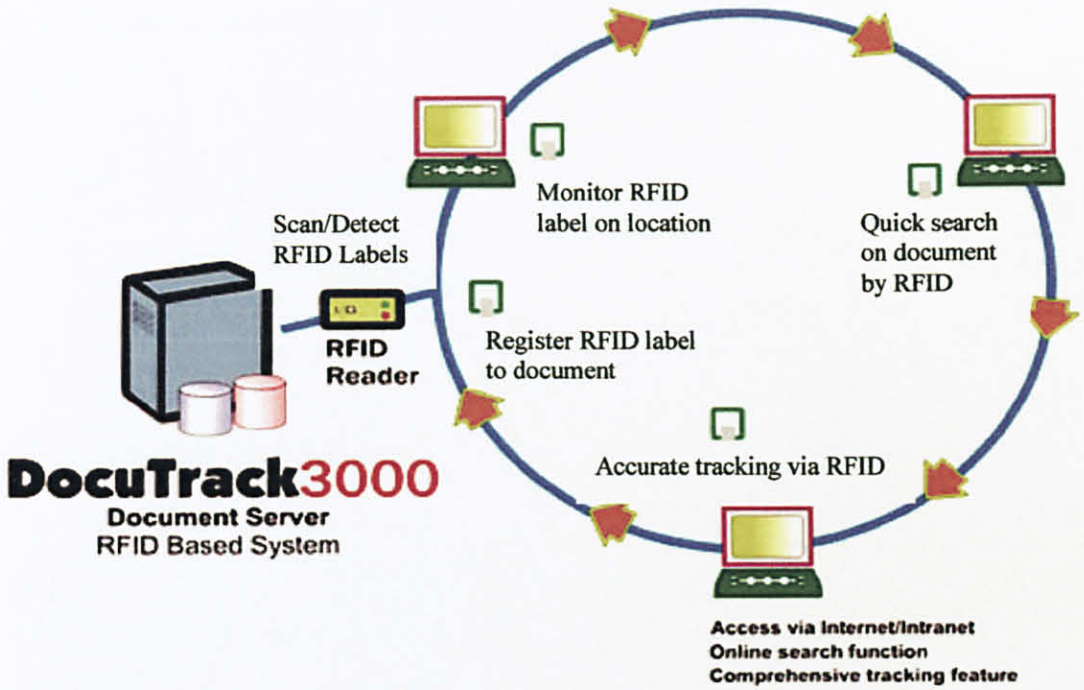


Figure 11: DocuTrack3000 system diagram

For the electronic documents, they will be encrypted and stored on a central media server. Printed documents and files are tracked using electronic tracking systems. Collaborative document development and workflow management tools enable users to develop, share, approve, and archive electronic documents in one streamlined system. The result is higher productivity and increased profit (DocuTrack3000 Intelligent Document Tracking, n.d).

CHAPTER 3
METHODOLOGY

3.1 Project Identification Procedure

There are several methodologies need to be done in order to complete the project. The Gantt chart of the project timeline can be found in Appendix A. Whereas the methodologies are summarized below:-

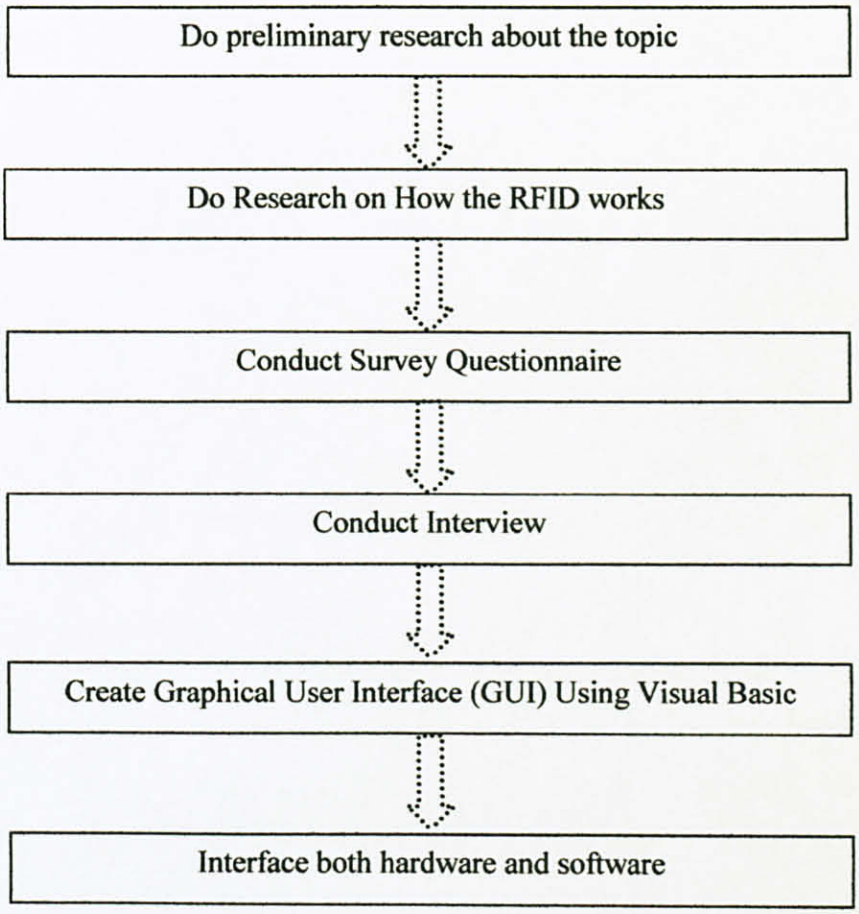


Figure 12: Project Flow Chart

3.1.1 Preliminary Research about the Topic.

In order to undergo the project, the basic understanding of the RFID technology is very important. The Dissertation, Paper works, journal, engineering books or anything relevant to the project are reviewed.

3.1.2 Do Research on How the RFID works

Instead of reading about the RFID technology, the basic understanding of how the RFID system works is very important. The author has gone to the communication lab to play around with the RFID kit. This is done in order to observe and understand the nature of RFID system itself.

3.1.3 Survey Questionnaire

A survey questionnaire has been conducted in order determine the feasibility of the project. A set of survey questionnaires has been created and randomly distributed to ninety four respondents. The survey was conducted through internet. The result of the survey is shown in the result and discussion within this report.

3.1.4 Interview

The author has conducted two interviews with the exam unit personnel regarding the UTP Strong room especially for the exam papers storage. These interviews were conducted in both verbally and non-verbal (through email) in order to understand more about what is actually stored in the room, the current technology used for the security of the room, how the document been stored and etc. The result of the interviews is shown in Appendix C.

3.1.5 Create *Graphical User Interface (GUI) Using Visual Basic.*

The author has decided to use Visual Basic 2008 to create an interface between tag and reader. In the project, this software is used to create a graphical user interface and linking information within the computer. The information on Visual Basic 2008 were obtained from various written sources, which includes lecture notes from UTP Business Information Technology students, as well as books that provide guidelines on how to use Visual Basic. Some development is shown result and discussion.

3.2 Applied Tools and Technology

3.2.1 Software

- MICROSOFT WORDS

  This software is used to complete job on documentation.

- VISUAL BASIC 2008 version

  This software is used to develop programme for the operation of the system.

- My SQL

  "My Structured Query Language" is a program that runs as a server providing multi-user access to a number of databases ("My SQL").

3.2.2 Hardware / Tools

Figure 13: *ActiveWave RFID Kit.*

3.2.2.1 RFID Kit Components:

- 1 Activewave Reader and Power Supply
- 1 Activewave Reader RS-232 Connector Cable
- 1 ActiveWave Reader RJ-45 Cable
- 6 Activewave Wristband Tags
- 2 ActiveWave Jumbo Tags
- 2 Activewave Card Tags
- 1 Programming Station Software Application
- 1 Tracker Program Software Application (Demo Version)
- 1 API with Documentation and Example Software Application

3.2.2.2 Specification

- **Tags** – Based on the comparison of active and passive tags; and also the operation frequency in the Literature Review, the author has decided to choose active tags with ultra high frequency (UHF) band which suits the best to be used throughout this project. Several ActiveWave tag model are available, each one transmitting at 916 MHz, 868 MHz or 927 MHz. Tags may be electronically enabled or disabled, so they can be "seen" or "unseen" by ActiveWave Readers. All ActiveWave tags have anti-collision circuitry that assures each tag's information is received when more than one tag is transmitting. An on-board temperature sensor can also be included to fit the customer's requirements.

- **Readers** – Readers interface the Host applications to the rest of the ActiveWave system. Readers transmit data at 433 MHz and receive data at 916 MHz, 868MHz or 927 MHz. Readers communicate to the Host computer via an RS-232 cable or via an Ethernet network connection. Readers are used to read the tags and transmit the received data to the Host computer. Readers are also used to enable, disable, wake up and program tags.

3.3 Project Overview

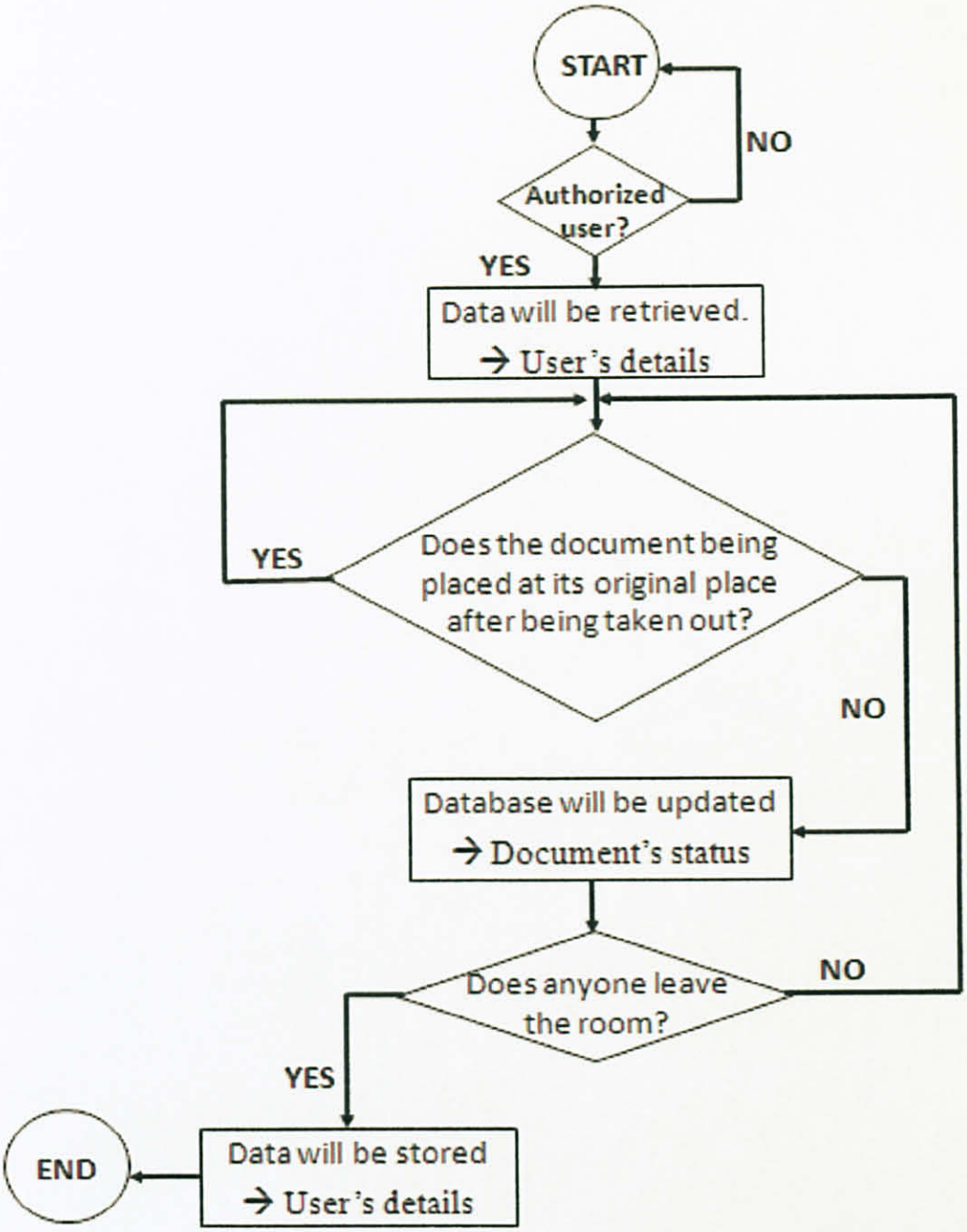


Figure 14: Tracking System Flow Chart

3.3.1 System Overview

Basically, the main components of the system are Active RFID tags, readers and controller PC. Two readers will be used for this project. One reader will be located near the cabinet to monitor the document folder in and out the cabinet. Another reader will be located at the entrance. The controller PC will be located in the room. Because of the document stored is high value, only the authorized staffs have access to the strong room and the controller PC. This controller PC contains the system software and control the operation of the entire system.

The system works by the sequence of process as follows;-

1. Each course has its own folder where the copies of the draft are kept until the final version; both soft and hard copies together with the answer scheme, course syllabus since all are subjected to audit at any time.
2. The RFID (radio frequency identification) tag is then attached to the folder.
3. The administrator will key-in the details in the database.
4. If an authorized user enters the room and takes out the folder, the time will be recorded and the person who takes it out will also be recorded.
5. The other users can view the history of the document in order to check the availability of the document and also to keep track where the document is and with whom.

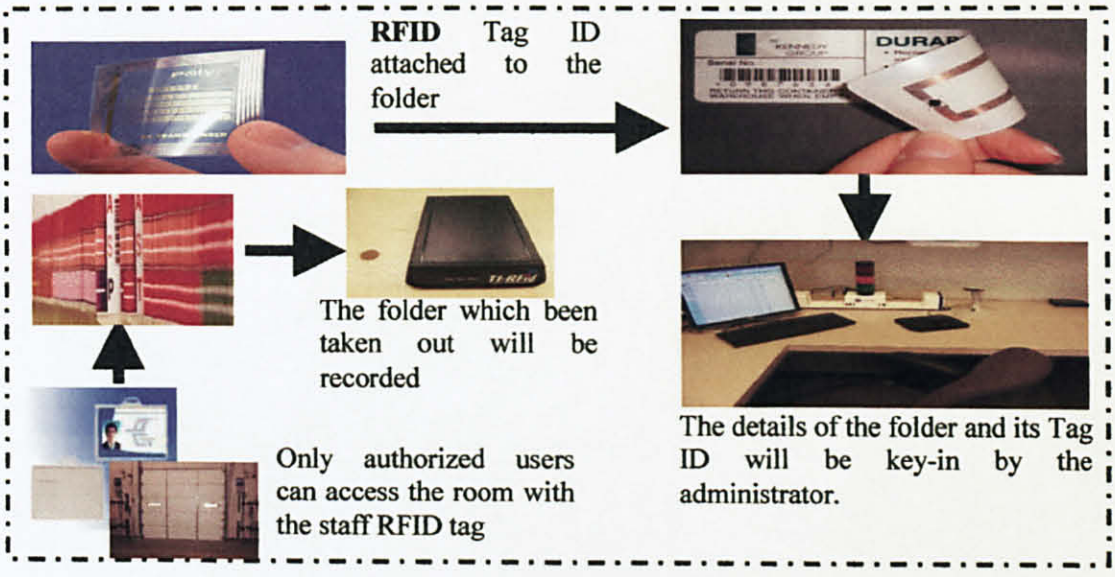


Figure 15: How the System Works

3.3.2 Authentication and Access Control Database Server

The database (My SQL) is created in order to store details about the users and document folders. Users' details such as user's identification number, password and full name are created, followed by the documents' details such as the course code, course name and location of the document. In order to increase the security of the system, the database will only be viewed by the administrator, who is responsible in editing the users or documents' details.

Then, the user interface is created in order to help the users to view the availability of the documents. The history of the system can be checked through the interface main page. Besides that, the data is retrieved from the database, as well as from the hardware. The details of the documents will be displayed based on the availability of the document.

CHAPTER 4
RESULTS AND DISCUSSION

4.1 Introduction

This chapter will basically discuss on the results obtain from the survey questionnaire; and the interfaces and databases which are created by both Visual Basic 2008 and My SQL software respectively.

4.2 Results for Survey Questionnaire

The author has conducted a general survey questionnaire via internet. The result for the questionnaire has been obtained as per attach in Appendix B. There are about ninety four respondents that have participated in this survey. Mostly, the respondents' ages are between 19 to 33 years old and coming from different technical background such as education, medical, finance and engineering. The results analysis of the survey is shown below:

1. Have you ever heard about RFID?

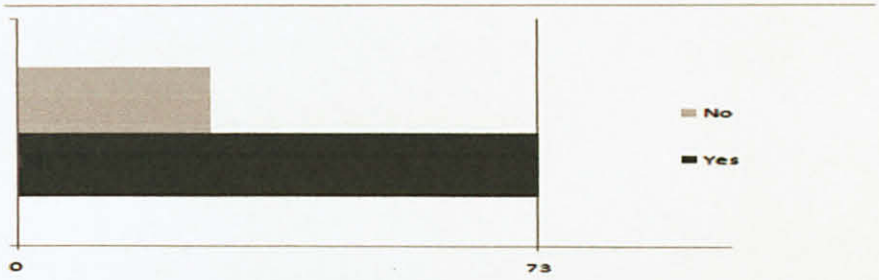


Figure 16: Bar Chart for Question 1

About 73 % respondents answer "Yes" and 27% answers "No". It shows that most of the respondents already know about the RFID technology.

2. Have you ever faced difficulties in searching for documents?

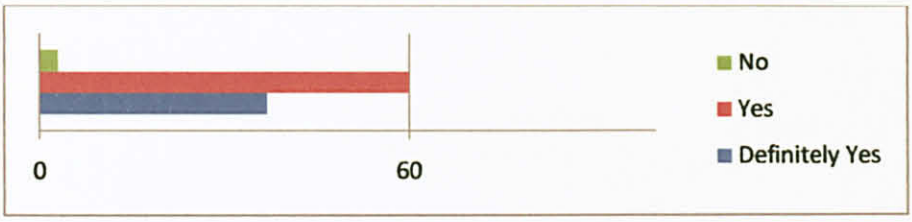


Figure 17: Bar Chart for Question 2

Only 3 % respondents answer "No". It shows that most of the respondent always facing difficulties in searching for documents.

3. Have you ever lost any important document?

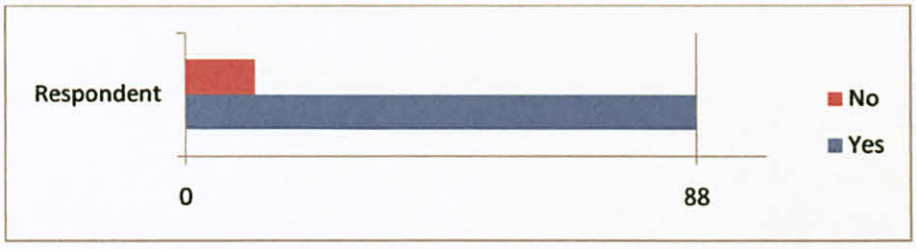


Figure 18: Bar Chart for Question 3

About 88 % respondents answer "Yes". It shows that most of them have experience in losing important document.

4. Does the existing method of documentation give you so much trouble?

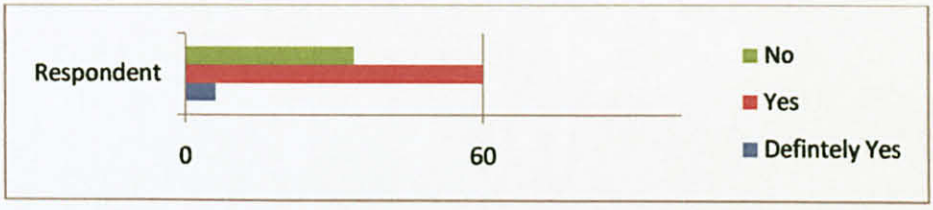


Figure 19: Bar Chart for Question 4

About 34 % respondents answer "No". It shows that most of the respondents have experience the troublesome from the existing documentation method.

5. Have you ever heard about application of RFID in document tracking system?

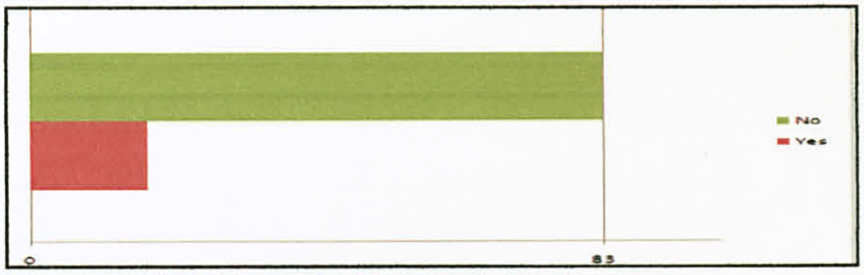


Figure 20: Bar Chart for Question 5

Only 17 % answer "Yes". It shows that they are not aware with the latest application of RFID in tracking document.

6. What are your expectations from the system? Please describe it.

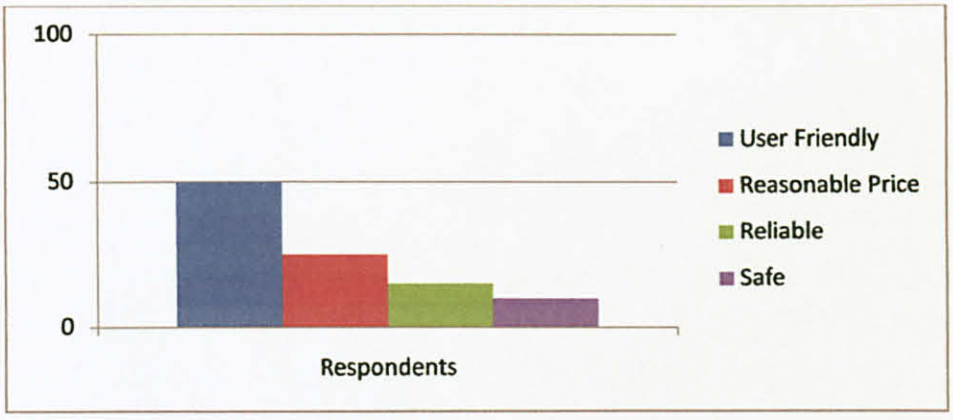


Figure 21: Bar Chart for Question 6

About 50 % of the respondents want the system would be "User friendly", 25% of them want the system will be sold with "Reasonable price". Besides that, 15% of the respondents want the system will be "Reliable" and the rest about 10% of them want the system to be secured in design.

4.3 Interfaces

4.3.1 Graphical User Interface (GUI) Using Visual Basic

The GUI for this system has been developed. Basically, an authorized user will be ask to enter the username and password in a prompt window as follow

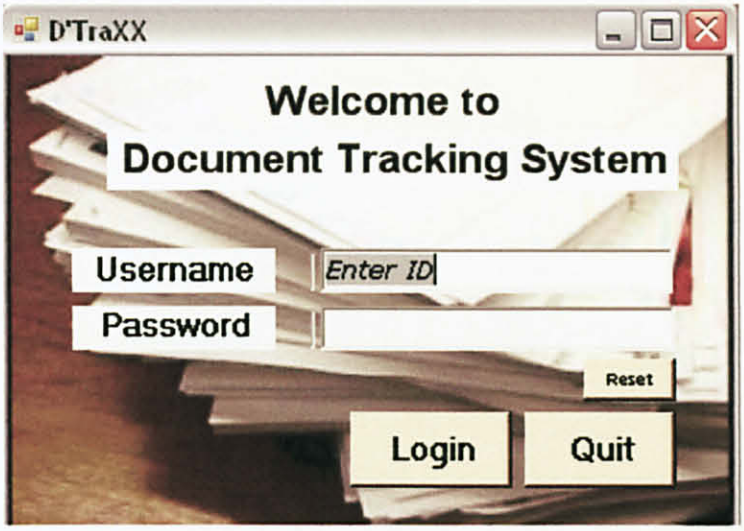


Figure 22: Login Window

After the authorized user successfully login to the system, the system will differentiate whether the user is the administrator or the common user. If he/she is the administrator, he/she will be brought to the *Administrator Main Menu* window as shown in Figure 23. Whereas, if he/she is a common user, a *User Main Menu* window will be popped up as shown in Figure 24. The system will differentiate the menu for both administrator and common user when any user log-in into the system. This will increase the security of the system itself. That means for the common user, he/she cannot change or edit anything to the systems. For the administrator, it is vice versa. He/she can edit the database of the document and user. He/she also can power on and off the RFID reader from the document tracking system. If compared to the common user, they only can view the *Document List* and *Search Document* as shown in Figure 24. They do not have the authority to power on and off the RFID system. This could increase the security.

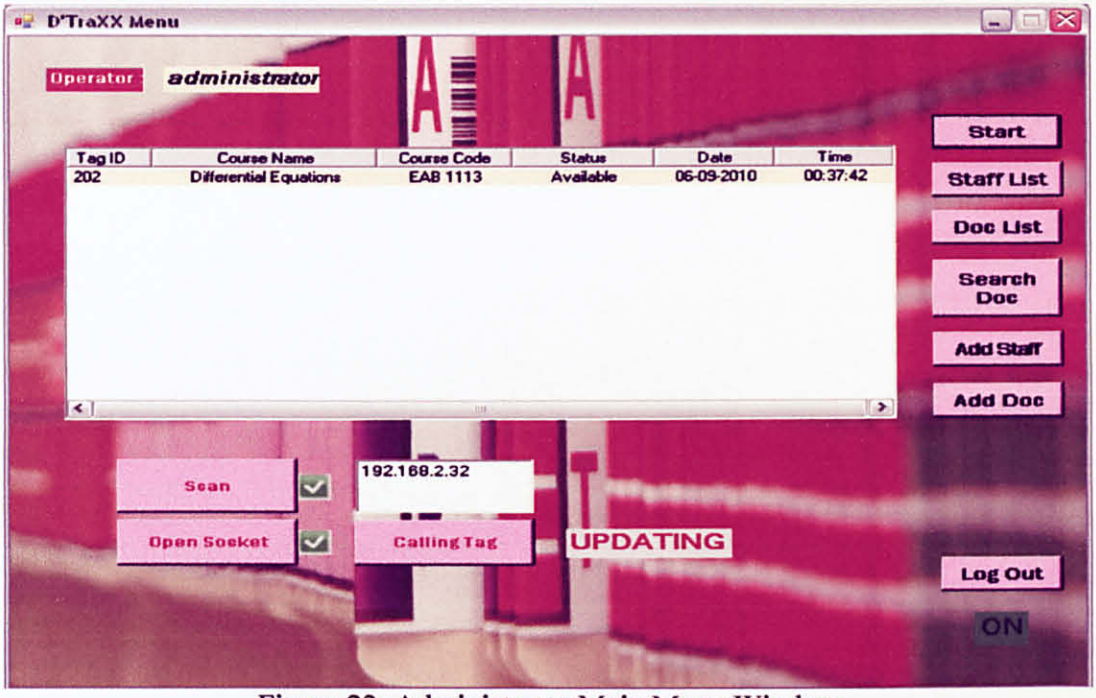


Figure 23: Administrator Main Menu Window

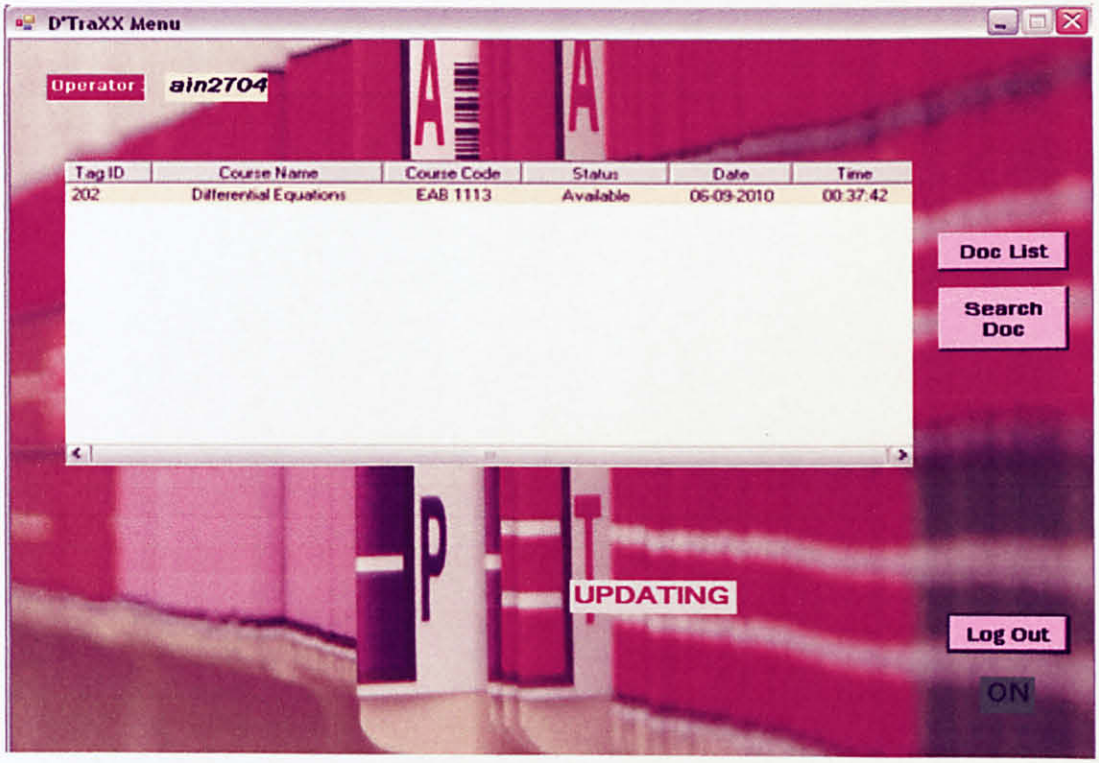


Figure 24: User Main Menu Window

From the *Administrator Main Menu* window, there are many buttons for the administrator to choose. There are *Scan* button, *Open Socket* button, *Calling Tag* button, *Start* button, *Add Staff* button, *Add Doc* button, *Search Doc* button, *Doc List* button, *Staff List* button and *Logout* button.

Firstly, to connect the DocTraXX 2.0 system to the reader, the administrator has to configure the IP of the reader. Then, click the *Scan* and *Open Socket* buttons to connect the system with the hardware. After that, he/she needs to click *Calling Tag* button to update the tags that available. Just click the *Start* button to display all the tags that available in the List view box. The indicator of the reader will change from *OFF* to *ON*. If he/she wants to log out from the system, he/she needs to click the *Logout* button from the Main Menu. It will not disturb the system even though he/she has logout. So, the next user can just login as usual. If he/she wants to find any document, just click *Search Doc* button as shown in figure 25 below.

Figure 25: Search Document Window

Besides that, in order to add new staff, firstly the administrator need to click *Add Staff* button from the Main Menu and the *Add New Staff* window will pop up as shown in Figure 26.

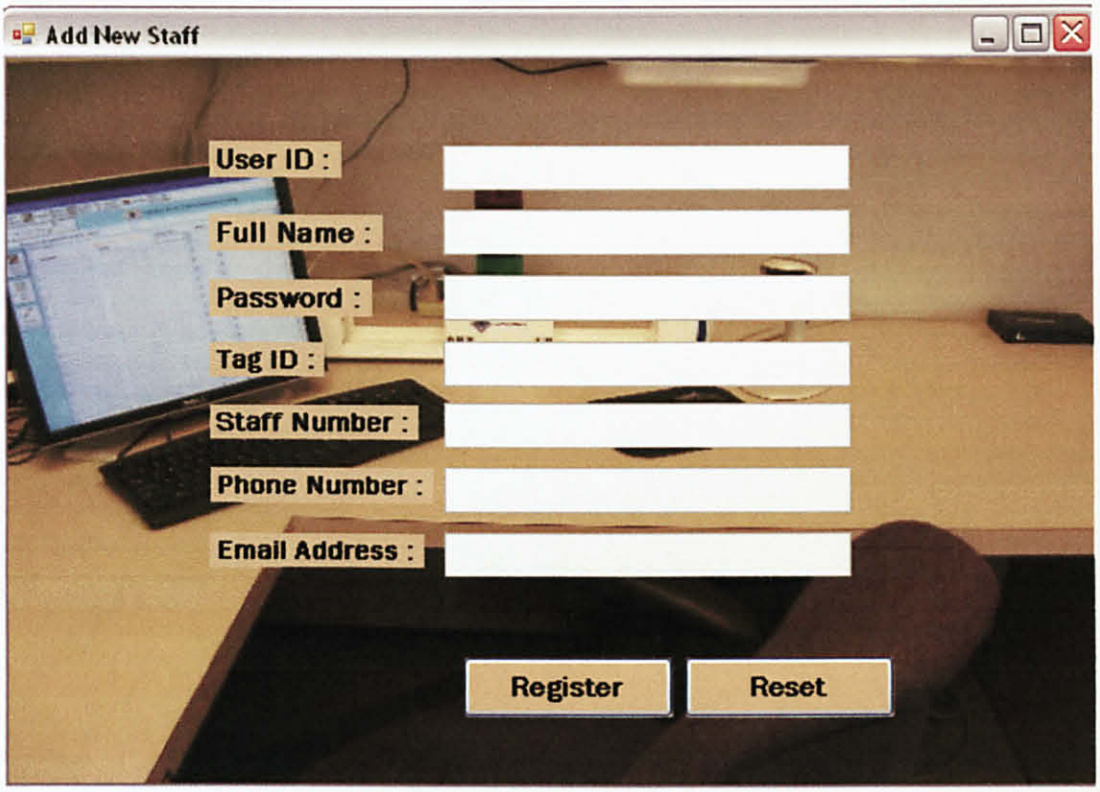


Figure 26: Add New Staff window

The administrator then will need to fill up all the details of the staff. He/she need to click *Register* button to register the staff in the database. Therefore, the staff will have an authority to access the system.

In order to add a new document into the database, the administrator also needs to do the same procedure as for the registration for the new staff. The *Add New Document* window will display as shown in Figure 27.

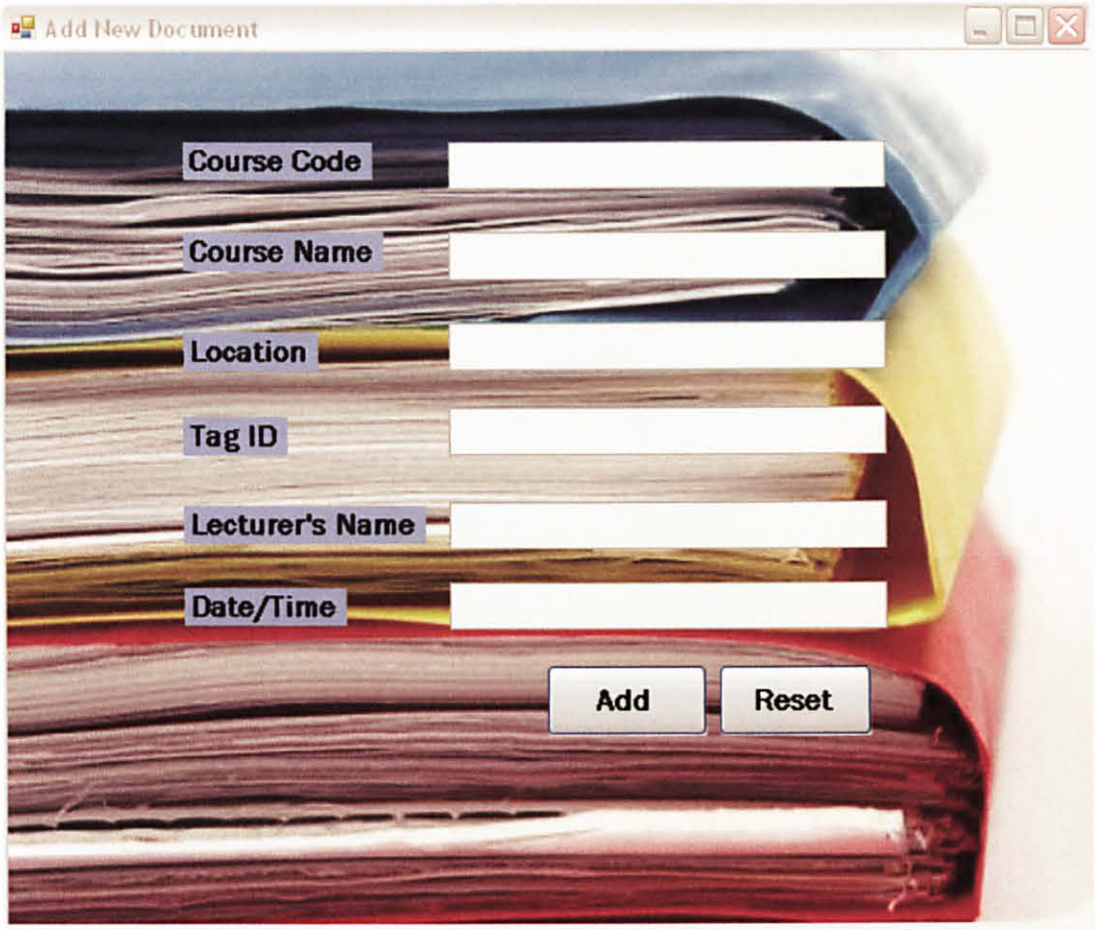


Figure 27: Add New Document window

The administrator/common user can view the document list by clicking the *Doc List* button as shown in Figure 28. It will display the details of the document. However, if he/she would like to view staff list, he/she may click the *Staff List* button as shown in figure 29. It will display the details information of the staff.

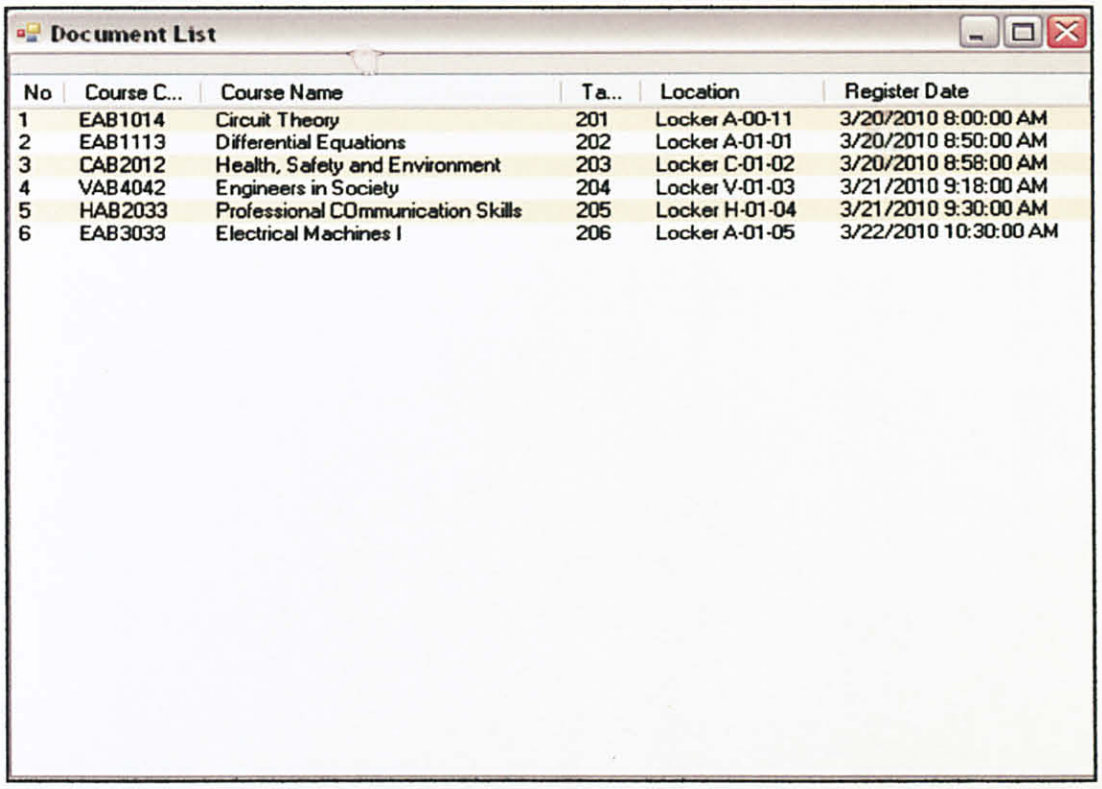


Figure 28: Document List window

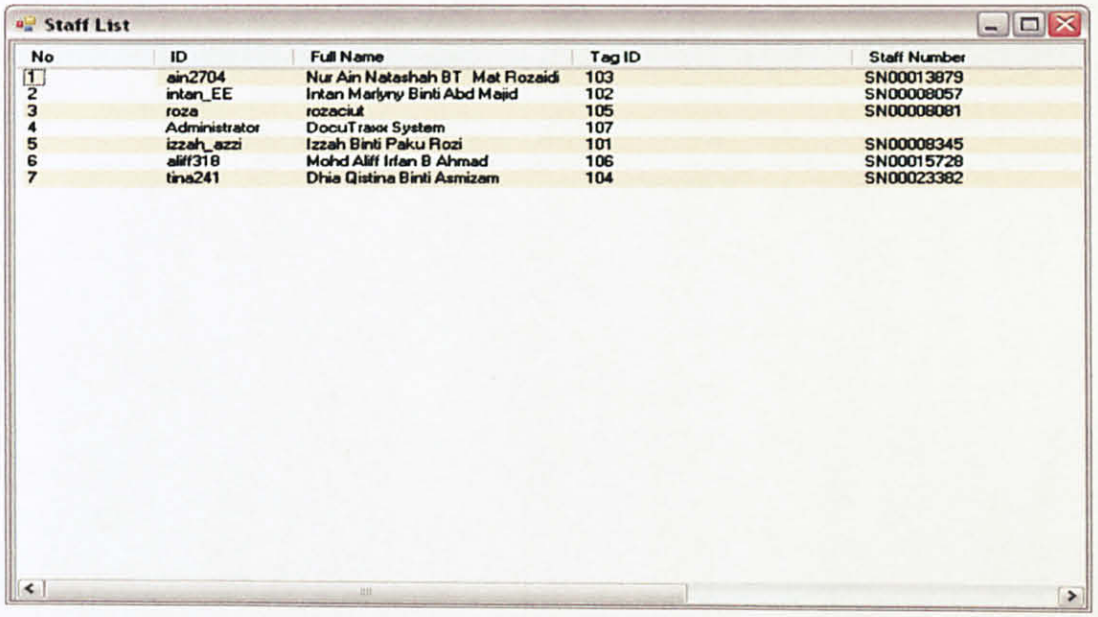


Figure 29: Staff List window

4.3.2 Database

The System database will be developed into two parts, which are documents and users' part.

4.3.2.1 User Database

As for the users part, user's identification number and the full name will be stored within database. Only users listed in the database are allowed to enter the document room, and access to any documents. However, in order to edit any changes to the user database, only the administrator can do it. The user database created as shown in Figure 30 below.

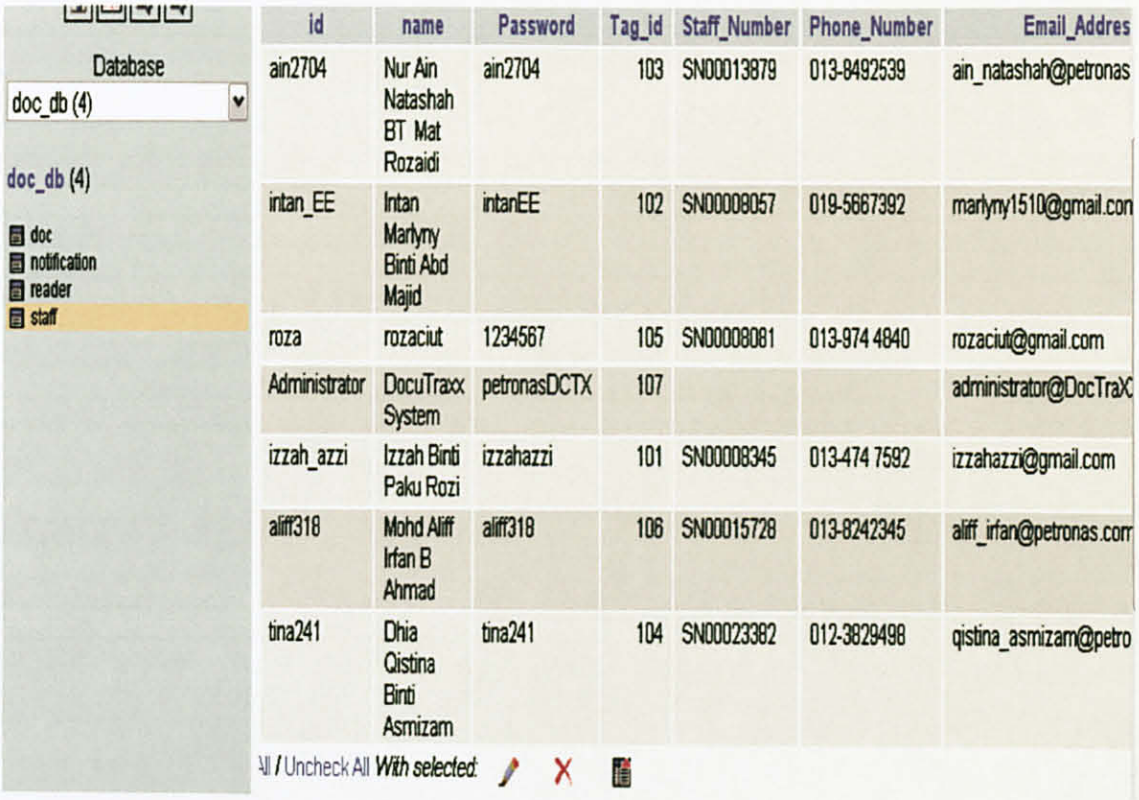


Figure 30: Database for user

4.3.2.1 Document Database

For documents part, details listed are the Course Code, Course Name, Document Tag ID and storage location. The document database created as shown in Figure 31 below.

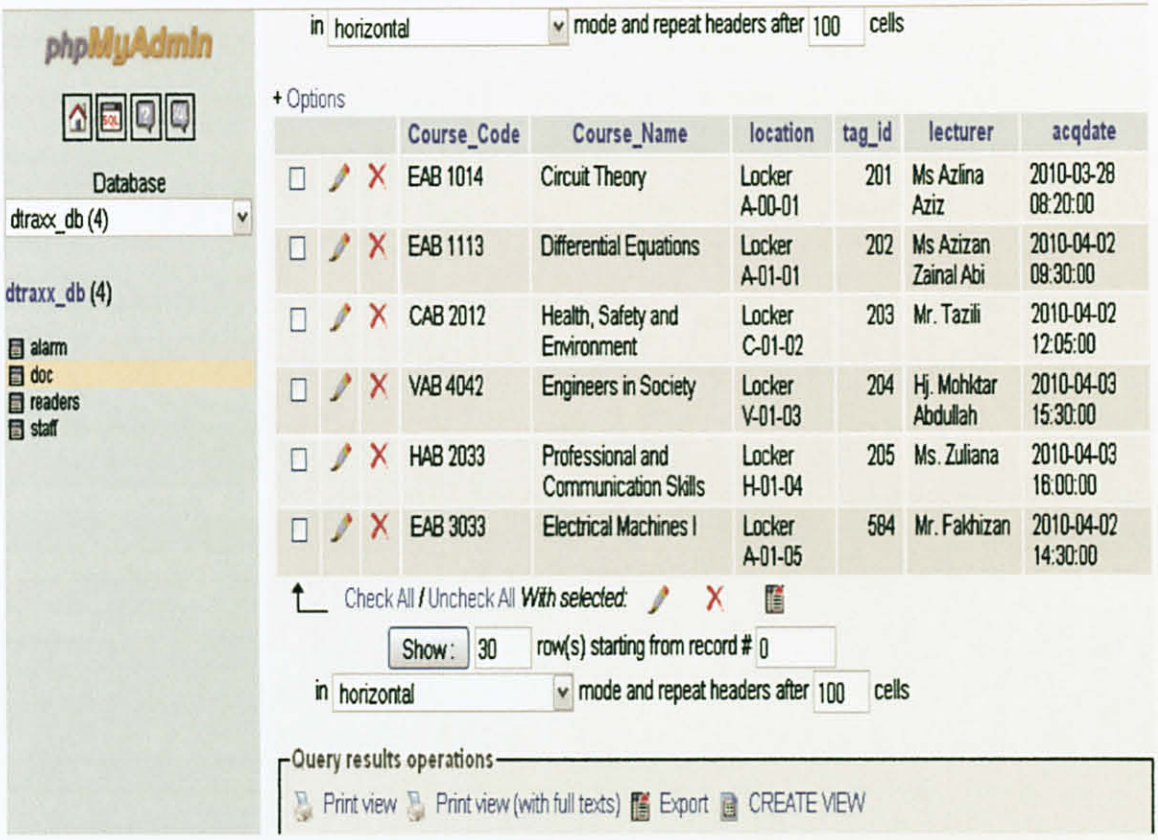


Figure 31: Database for documents

4.4 Discussion

4.4.1 Result for Survey Questionnaire

From the result of the survey, we can see that most of the respondents facing difficulties in managing their document. Most of them think that finance institution will face the most difficulties in handling stack of the documents. About 66% respondents agree that the existing method of documentation will give them so much trouble. Most of them also very aware about the current technology like barcode and RFID. However, most of them still do not know that RFID could be used in tracking document. For the conclusion, most of the respondents agree the needs of a new secure technology to be invented in order to help them manage their documents.

4.4.2 Graphical User Interface (GUI) Using Visual Basic.

Most of the GUI has been developed using Visual Basic 2008. The author tries to make the GUI as simple as possible so that the new user can adapt easily with the interface. There are number of problems encountered throughout completing the project. Firstly prior to the coding stage of the project in order to detail understand the Visual Basic software. Secondly, the coding needs to be developed in order to interface the software created and the hardware. Therefore, in order to overcome the problems encountered, the author has learnt herself on the basic of Visual Basic 2008 software by referring to the relevant textbooks available in the library. She then finally has managed to complete the project by getting advice and consultation from the expert one in the programming language. However, although the GUI is finalized, the author still can improve it to meet client requirement.

CHAPTER 5
CONCLUSION AND RECOMMENDATION

5.1 Conclusion

The application of RFID technology in document tracking has been identified through three means which are literature research, questionnaire surveys, and interviews. By using the preliminary data from those methodologies, the author has managed to complete this project which is entitled the "Document Tracking System Using RFID (D'TraXX)". This system is especially designed to be implemented for UTP strong room which documents stored are very high in value. The system will help in improving the security of the current document management of UTP Strong Room. Furthermore, the movement of documents can be traced in order to prevent misfiled, misplaced or stolen.

5.2 Recommendation

By adding new features, it will benefit the system for better advancement. The recommendations are as follows:

i. Improve the security of the Strong Room entrance by combining RFID with Biometric Technology such as fingerprints, iris scan, hand recognition and etc.

ii. The system security within the room can also be increased by combining current RFID system with CCTV. Thus the efficiency and reliability of current system can be improved.

REFERENCES

Ann Cavoukian, PhD, Information and Privacy Commissioner of Ontario. (2008). Journal: *Guidance for Health-Care Providers : RFID and Privacy*

Association for Automatic Identification and Mobility (AIM). Article on *Technologies: RFID / What is RFID / Component.* Retrieved on July 16, 2009 from http://www.aimglobal.org/technologies/RFID/RFID_Components.asp

Daniel M. Dobkin, (2005, October). Article on The RF in RFID: *Physical layer operation of passive UHF tags and readers.* Retrieved on October 2, 2009 from http://www.enigmatic-consulting.com/Communications_articles/RFID/RFID_frequencies.html

DocuTrack3000 Intelligent Document Tracking. Retrieved on September 1, 2009 from http://www.docutrack.biz/docutrack/ .

How RFID "Works". Retrieved on January 14, 2010 from http://www.rfidtagsource.com/how-rfid-works

Ken Cheung (2007). Article on *GAORFID Introduces LocateWare RFID Middleware for Locating, Tracking RFID Asset Tracking.* Retrieved on August 21, from http://edageek.com/2007/09/21/gaorfid-locateware/

Łukasz Geldner & Paweł Nowiński.(2003). Bachelor of Science Thesis: *RFID Reader/Writer.*

National Institute of Standards and Technology (2007): Journal: Guidelines for Securing Radio Frequency Identification (RFID) Systems

RFID Applications: Document Tracking (2006). Retrieved on July 20, 2009 from http://www.rfidlowdown.com/2006/10/rfid_applicatio_1.html

Siti Ruzzana Binti Roslant. (2009). Dissertation: *Document Tracking using Radio Frequency Identification (RFID)*

White Paper. (2009). Journal: Part 1: Active and Passive RFID: *Two Distinct, But Complementary, Technologies for Real-Time Supply Chain Visibility*

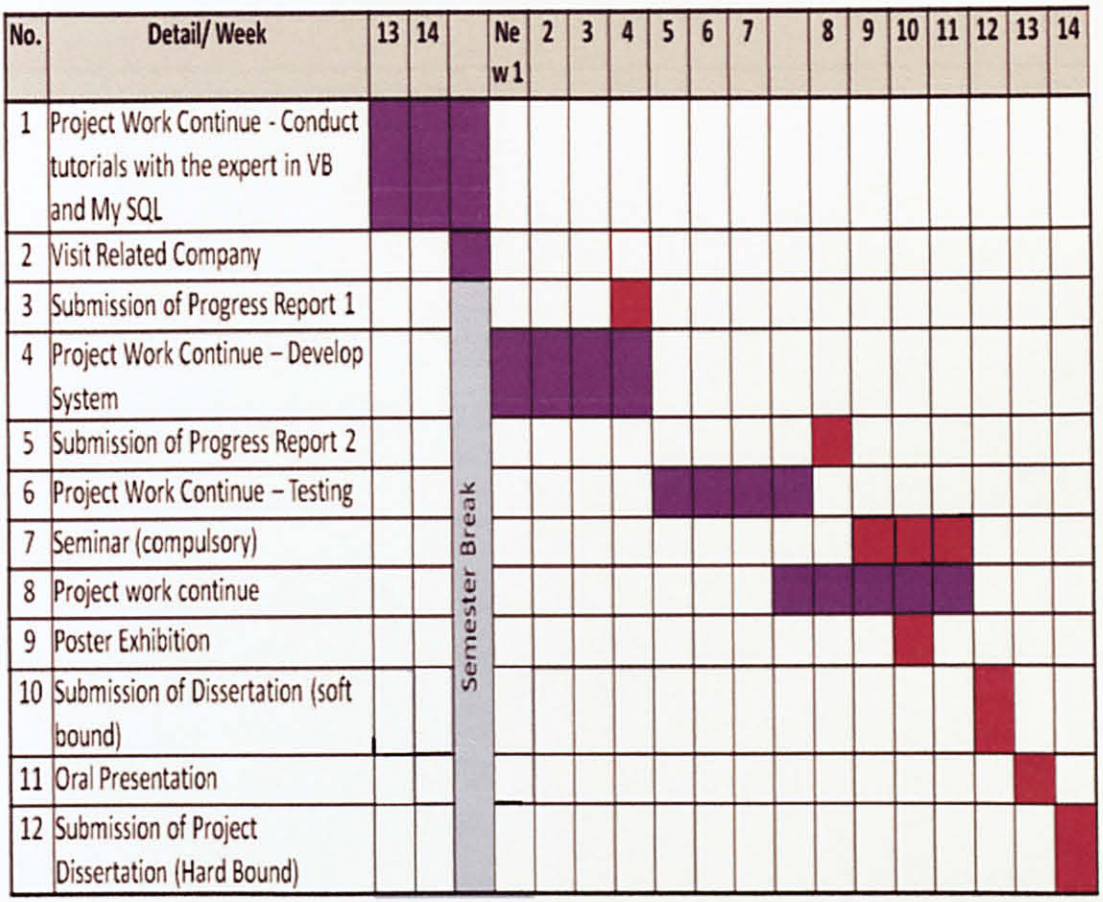
White Paper. (2009). Journal: *Solving the New Technology Requirements for RFID Business Applications: How Laser Printing Technologies are Adding Value To an Evolving Business Landscape.*

Wikipedia, the free encyclopaedia. Article on *Radio-Frequency Identification.* Retrieved on September 1, 2009 from http://en.wikipedia.org/wiki/Radio-frequency_identification

APPENDICES

APPENDIX A:

PROJECT GANTT CHART



Gantt Chart for the Second Semester of Final Year Project

APPENDIX B:

SURVEY QUESTIONNAIRES RESULT

	Have you ever heard about RFID?	Radio Frequency Identification (RFID) is technology similar in theory to bar code identification.	Have you ever faced difficulties in searching for documents?	Have you ever lost any important document?	Does the existing method of documentation give you so much trouble?	Which institution do you think would face the most difficulties in handling the stack of documents?	Have you ever heard about application of RFID in document tracking system?	What are your expectation from the system?Please describe it.
0:49:17	No	RFID	Yes	Yes	No	Finance	No	
0:53:24	Yes	RFID	Yes	No	No	Legal	No	
0:53:45	Yes	RFID	Definitely yes.	Yes	Yes	Finance	No	user friendly
0:53:56	No		Yes	Yes	Yes	Medical	No	easy to use, efficient, xmudah rosak, high durability
0:54:43	No	Barcode	No	Yes	No	Oil and Gas	No	
0:58:35	Yes	RFID	Definitely yes.	Yes	Yes	Finance, Legal	No	less cost
0:58:04	Yes	RFID	Definitely yes.	Yes	Definitely yes.	Academic, Finance, Legal, Medical, Oil and Gas	No	should be user friendly, no wires, no hassle, not expensive, affordable, reliable and easy to maintain, easy to set up, easy to remove. thank you
0:59:10	No	RFID	Definitely yes.	Yes	Yes	Academic, Finance, Legal, Medical, Oil and Gas	No	Dont know anything about it. dont have idea what suolhd i expect from it. maafin
1:00:59	Yes	RFID	Yes		Yes	Academic	No	
1:02:30	Yes	RFID	Definitely yes.	Yes	No	Academic, Finance, Legal, Medical, Oil and Gas	No	User friendly, reliable.
1:02:46	Yes	RFID	Yes	Yes	Yes	Finance	No	good idea.. ease us.. easy to track the document.. save time.. tq
1:03:14	Yes	RFID	Yes	Yes	Yes	Finance	Yes	flexible
1:05:49	No	RFID	Yes	Yes	No	Legal	No	mendalah apa ni ? emmm.... btw... all the best RFID for all ... PAS for ALL
1:07:09	No	Barcode	Definitely yes.	Yes	Yes	Finance	No	
1:14:43	Yes	Barcode	Definitely yes.	Yes	No	Finance	No	
1:16:34	Yes	RFID	Yes	Yes	Yes	Finance	Yes	Security.. dll
1:18:30	No	RFID	Yes	No	Yes	Oil and Gas	No	
1:20:51	Yes	RFID	Definitely yes.	No	No	Finance	No	
1:24:33	Yes	RFID	No	No	No	Legal	No	
1:25:22	Yes	RFID	Definitely yes.	Yes	No	Medical	No	
1:26:22	Yes	Barcode	Yes	Yes	Yes	Academic, Finance, Legal	No	user friendly, cheap,effective, last long
1:29:34	Yes	RFID	Yes	Yes	Yes	Finance	No	Easy understanding, easy learning
1:32:05	Yes	RFID	Yes	Yes	Yes	Legal	No	
1:34:12	Yes	RFID	Definitely yes.	Yes	Yes	Finance	No	
1:37:35	Yes	RFID	Yes	Yes	Yes	Academic, Finance, Legal, Medical, Oil and Gas	Yes	
1:37:53	Yes	RFID	Definitely yes.	Yes	Definitely yes.	Academic	Yes	
1:41:50	Yes	Barcode	Yes	Yes	No	Academic	No	
1:42:25	No	Barcode	Yes	Yes	Yes	Finance, Legal, Medical	No	can help users in their problems..
1:51:50	Yes	RFID	Definitely yes.	No	No	Academic	No	applicable
1:53:42	Yes	RFID	Yes	Yes	Yes	Legal		user friendly
1:53:50	Yes	RFID	Definitely yes.	Yes	Yes	Finance, Legal	No	Definitely user friendly-enable new user to master the system without the need to go for training etc cheap-doesn't cost much so that can be installed and used for personal usage safe-information stored cannot be accessed by illegal partics

	Have you ever heard about RFID?	Radio Frequency Identification (RFID) is technology similar in theory to bar code identification.	Have you ever faced difficulties in searching for documents?	Have you ever lost any important document?	Does the existing method of documentation give you so much trouble?	Which institution do you think would face the most difficulties in handling the stack of documents?	Have you ever heard about application of RFID in document tracking system?	What are your expectation from the system?Please describe it.
10 2:00:13	Yes	Barcode	Yes	Yes	Yes	Academic	No	This thing maybe helpful to find lost document because it can detect it within its radius of coverage.
10 2:03:10	Yes	RFID	Yes	Yes	No	Finance	No	Used by any level of individuals, not necessarily high end institutions only.
10 2:05:42	No		Yes	Yes	Yes	Finance	No	not pretty sure about it..
10 2:09:47	Yes	RFID	Definitely yes.	Yes	Yes	Finance	No	help me to track the document easily..
10 2:10:29	Yes	Barcode	Definitely yes.	Yes	Yes	Academic, Legal	No	
10 2:21:25	No	Barcode	Yes	Yes	Yes	Finance, Medical	No	user friendly
10 2:27:45	Yes	RFID	Definitely yes.	Yes	No	Academic	Yes	
10 2:29:52	No	RFID	Yes	Yes	No	Finance	No	
10 2:55:23	No	Barcode	Yes	Yes	Yes	Academic	No	
10 2:59:22	No	Barcode	Yes	Yes	Yes	Academic	No	
10 3:03:10	Yes	RFID	Yes	Yes	Yes	Academic, Finance, Legal, Oil and Gas	No	i can find my interested documents easily
10 3:22:25	Yes	RFID	Yes	Yes	Yes	Finance	No	neat and tiday
10 4:05:39	Yes	RFID	Yes	No	Yes	Academic, Finance, Legal, Medical, Oil and Gas	No	is it the same as inside IRC book. so,its need for less maintenance. need really good planning to create the system. especially when using in library because library is one hell pack of documents.
10 5:59:32	Yes	RFID	Yes	Yes	Yes	Finance	Yes	user friendly
10 6:59:54	Yes	RFID	Yes	Yes	Yes	Academic, Finance, Medical, Oil and Gas	No	
10 7:00:09	Yes	Barcode	Yes	Yes	No	Finance, Medical	No	- waste time help people
10 7:06:35	No	RFID	Definitely yes.	Yes	No	Academic, Finance, Legal, Medical, Oil and Gas	No	
10 7:07:32	Yes	RFID	Definitely yes.	Yes	Yes	Medical	No	Help me
10 7:10:32	Yes	RFID	No	Yes	No	Academic, Finance, Oil and Gas	Yes	able to commercialize
10 7:21:44	No	RFID	Yes	Yes	Yes	Academic	No	
10 7:25:07	Yes	RFID	Definitely yes.	Yes	No	Finance, Medical, Oil and Gas	No	Easy to use
10 7:36:45	Yes	RFID	Yes	Yes	Yes	Medical, Oil and Gas	No	simple and easy
10 7:45:44	Yes	Barcode	Yes	Yes	Yes	Medical	No	Simple and reliable
10 7:49:00	Yes	RFID	Yes	Yes	No	Finance	No	
10 7:54:55	No	Barcode	Yes	Yes	Yes	Academic, Medical	No	
10 7:59:45	Yes	RFID	Definitely yes.	Yes	No	Academic, Finance, Legal, Medical	Yes	
10 8:19:16	Yes	RFID	Definitely yes.	Yes	Yes	Academic, Finance, Legal	No	I am not sure what is this system and how it works and also what does it do. Generally i would want an universally easy system: 1) can be accessed anywhere there is internet 2) good GUI 3) fast, no lag
10 8:20:38	No	Barcode	Yes	Yes	Yes	Academic, Finance, Legal, Medical, Oil and Gas	No	Can detect the document fastly, even in the cupboard
10 8:21:39	Yes	RFID	Definitely yes.	Yes	Yes	Academic	No	
10 8:35:11	Yes	RFID	Definitely yes.	Yes	No	Academic, Finance, Medical	Yes	able to understand the command of the user and definitely be user friendly

	Have you ever heard about RFID?	Radio Frequency Identification (RFID) is technology similar in theory to bar code identification.	Have you ever faced difficulties in searching for documents?	Have you ever lost any important document?	Does the existing method of documentation give you so much trouble?	Which institution do you think would face the most difficulties in handling the stack of documents?	Have you ever heard about application of RFID in document tracking system?	What are your expectation from the system?Please describe it.
9:05:08	Yes	RFID	Yes	Yes	Yes	Legal, Oil and Gas	Yes	RFID must combine with good system interface and good SOP.
9:26:15	Yes	RFID	Definitely yes.	Yes	Yes	Academic, Finance, Legal, Medical, Oil and Gas	No	
9:45:42	Yes	Barcode	Definitely yes.	Yes	Yes	Finance	Yes	
9:57:43	Yes	Barcode	Yes	Yes	Yes	Legal	No	fast and reliable
10:21:04	Yes	RFID	Definitely yes.	Yes	Yes	Finance	Yes	
10:23:49	Yes	Barcode	Yes	Yes	No	Academic, Finance, Legal, Medical, Oil and Gas	No	
10:25:31	No	Barcode	Yes	Yes	Yes	Finance	No	
10:34:40	Yes	RFID	Definitely yes.	Yes		Finance	No	
10:52:36	Yes	RFID	Definitely yes.	No	Definitely yes.	Academic, Finance, Oil and Gas	No	Of course it would be user friendly, besides that, it should be easily to handle and does not waste people's time in searching documents.
10:52:43	Yes	Barcode	Definitely yes.	Yes	No	Finance	No	
11:57:23	Yes	RFID	Yes	Yes	No	Academic, Finance	No	
12:13:55	Yes	RFID	Yes	Yes	Yes	Academic	No	User friendly, cost economic portable
13:40:53	Yes	RFID	Yes	No	No	Finance, Legal	No	
13:56:07	Yes	Barcode	Yes	Yes	No	Finance	No	
14:18:46	Yes	RFID	Definitely yes.	Yes	Definitely yes.	Academic, Finance, Legal, Medical, Oil and Gas	Yes	user friendly, portable, easier maintenance & support
14:30:36	Yes	RFID	Yes	Yes	Definitely yes.	Finance	Yes	
15:09:23	Yes	RFID	Yes	Yes	Yes	Finance	No	user friendly
15:21:22	Yes	RFID	Yes	Yes	Yes	Academic, Finance, Legal, Medical	No	user friendly economic environmental friendly portable light and easy to install
15:30:49	Yes	RFID	Definitely yes.	Yes	No	Finance	No	
16:51:31	Yes	RFID	Yes	Yes	Yes	Academic	No	1.user friendly. 2.save time
18:15:07	Yes	RFID	Yes	Yes	No	Academic, Finance, Legal	No	
20:05:23	Yes	RFID	Yes	No	No	Academic, Finance, Legal, Medical	Yes	User friendly
20:07:57	No	Barcode	Yes	Yes	No	Finance, Legal, Medical	No	make the work easy
22:48:44	No	Barcode	Definitely yes.	Yes	Yes	Academic, Finance, Legal	No	user frenly
7:03:55	No	Barcode	Yes	Yes	Yes	Academic, Finance, Medical, Oil and Gas	No	
11:13:08	No	RFID	Yes	Yes	Yes	Academic, Finance, Legal, Medical, Oil and Gas	No	nano technology
12:38:28	Yes	RFID	Yes	Yes	Yes	Academic, Finance, Legal, Medical	No	
22:44:17	No		Yes	No	No	Legal	No	effective, simple operation and can ease the handling of the documents.. and of course user friendly too
12:27:43	No	Barcode	Yes	Yes	Yes	Academic, Finance, Medical, Oil and Gas	No	sorry, no idea on RFID. TQ.
16:18:24	Yes	RFID	Definitely yes.	No	No	Finance	No	

	Have you ever heard about RFID?	Radio Frequency Identification (RFID) is technology similar in theory to bar code identification.	Have you ever faced difficulties in searching for documents?	Have you ever lost any important document?	Does the existing method of documentation give you so much trouble?	Which institution do you think would face the most difficulties in handling the stack of documents?	Have you ever heard about application of RFID in document tracking system?	What are your expectation from the system?Please describe it.
11:10:04	Yes	RFID	Definitely yes.	Yes	Yes	Academic, Finance, Medical, Oil and Gas	Yes	reliable
11:43:01	No	Barcode	Definitely yes.	Yes	Definitely yes.	Finance	No	must be able to provide the fastest and effecient in tracking certain document and able to track back the lost document.
:07:09	Yes	RFID	Yes	No	Yes	Finance, Legal, Oil and Gas	Yes	

APPENDIX C:

INTERVIEW RESULT

---

* Required

1. Who is responsible to store the exam paper question in the strong room? * -The personnel, unit, or others :
   Exam unit staff only (all staff are required to read an oath of Secrecy)

2. How the lecturers give the final exam questions to the exam unit? * -Softcopy, hardcopy, both, cd, file or others :
   Softcopy(thumb drive) and hardcopy but exam unit staff will not have accessed to the softcopy until the final printing.

3. How the exam papers been stored in the strong room? * -Keep n A4 envelopes, computer(softcopy) or others :
   Each course has its own folder where we keep the copies of draft until the final version(soft&hard copies) together with the answer scheme, course syllabus since all are subjected to audit at any time.

4. Is there any shelf, cupboard or locker for storing the document? * Store in computer(softcopy) or others~~(Hardcopy) or both in the same room~~. :
   Yes, it is stored in a cabinet according to program respectively.

5. How the document inventory is being managed? * By coursecode/ programme/others :

   By program, course code, examiner's name.

6. If there is locker/shelf/cupboard, is it been locked and only the authorized user has the key? * or only the strong room is been lock, not the locker/shelf :

   All access to the document will have key / password.

7. Who has the access to the strong room? * :

   Examination unit executives

8. What type of the technology currently used for the security of the strong room? * Magnetic Contact(like door alarm system need password to enter or just swap magnetic card) or other?

   Traditional locked steel door using key and password.

APPENDIX C

INTERFACES CODING

Login

```
Imports System
Imports System.ComponentModel
Imports System.Threading
Imports System.Windows.Forms
Imports System.Runtime.InteropServices
Imports Microsoft.VisualBasic
Imports AW_API_NET
Imports System.IO
Imports MySql.Data.MySqlClient


Public Class DT_Intro

    Public ofName As String

    Private Sub officer_Load(ByVal sender As System.Object, ByVal e
As System.EventArgs) Handles MyBase.Load

    End Sub

    Private Function checkPswd(ByVal name As String, ByVal pswd As
String) As Boolean

        Dim found As Boolean = False
        Dim rows As Integer = 0
        Dim myConnString As String = "Database=dtraxx_db;Data
Source=localhost;User Id=root;Password="
        Dim cn As New MySqlConnection(myConnString)
        Dim cmdGH As New MySqlCommand("SELECT * FROM staff WHERE id =
'" & name & "'AND Password = '" & pswd & "'", cn)
        Dim daGH As New MySqlDataAdapter
        Dim dsGH As New DataSet
        Dim dtGH As New DataTable
        Dim offName As String = ""

        cn.Open()
```

```
Try
            daGH.SelectCommand = cmdGH
            daGH.Fill(dsGH, "employees")
            dtGH = dsGH.Tables("employees")
            rows = dtGH.Rows.Count()
        Catch ex As Exception
            MsgBox("Error: " & ex.Source & ": " & ex.Message,
MsgBoxStyle.OkOnly, "Connection Error !!")
        End Try

        If ConnectionState.Open Then
            cn.Close()
        End If

        If (rows > 0) Then
            found = True
            Dim namCmd As New MySqlCommand("SELECT name FROM staff
WHERE id = '" & name & "'", cn)
            Dim rdr As MySqlDataReader
            cn.Open()
            rdr = namCmd.ExecuteReader
            While rdr.Read
                offName = rdr.Item("name").ToString
            End While
            rdr.Close()
            cn.Close()
            ofName = offName
        Else
            found = False
        End If

        Return found

    End Function


    Private Sub LogoPictureBox_Click(ByVal sender As System.Object,
ByVal e As System.EventArgs)

    End Sub


    Private Sub lbl_Password_Click(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles lbl_Password.Click

    End Sub


    Private Sub Button1_Click_1(ByVal sender As System.Object, ByVal
e As System.EventArgs) Handles Button1.Click
        Me.Close()
    End Sub
```

```
    Private Sub btn_Submit_Click_1(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles btn_Submit.Click
        Dim userName As String
        Dim pwd As String

        userName = Me.TxtName.Text
        pwd = Me.TxtPswd.Text
        Form1.Label18.Text = userName
        If (checkPswd(Me.TxtName.Text, Me.TxtPswd.Text) = True) Then
            If userName <> "Administrator" And pwd <> "12345" Then
                Form1.BtnStart.Visible = False
                Form1.rfScanButton.Visible = False
                Form1.QueryButton.Visible = False
                Form1.BtnAddInmate.Visible = False
                Form1.BtnAddOfficer.Visible = False
                Form1.BtnOpList.Visible = False
                Form1.BtnStart.Visible = False
                Form1.IPListBox.Visible = False
                Form1.rfOpenSocketButton.Visible = False
                Form1.PictureBox1.Visible = False
                Form1.PictureBox2.Visible = False
                Form1.Show()
                Me.Hide()
            Else

                Form1.Show()
                Me.Hide()
            End If
        Else
            MessageBox.Show("Invalid ID Number or Password")
        End If

    End Sub

    Private Sub TxtName_TextChanged(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles TxtName.TextChanged

    End Sub

    Private Sub Button2_Click(ByVal sender As System.Object, ByVal e
As System.EventArgs)
        TxtName.Text = ("")
        TxtPswd.Text = ("")

    End Sub

    Private Sub Button2_Click_1(ByVal sender As System.Object, ByVal
e As System.EventArgs) Handles Button2.Click
        TxtName.Text = ("")
        TxtPswd.Text = ("")
    End Sub
End Class
```

Main Menu

```
Imports System
Imports System.Runtime.InteropServices
Imports Microsoft.VisualBasic
Imports AW_API_NET

Imports System.ComponentModel
Imports System.Threading
Imports System.Windows.Forms


Imports System.IO
Imports MySql.Data.MySqlClient

Public Class Form1
    Inherits System.Windows.Forms.Form
    Dim strhttp As String
    Dim ipIdx As Integer = 0
    Dim pubReaderID As Integer
    Dim c As Integer
    Dim txtSender As String
    Dim tagID As ListViewItem
    Private Strt As System.Threading.Thread
    Dim selID As String
    Dim Hconn As IntPtr
    Dim readerIP(20) As Byte
    Dim readerPort As UInt16
    Dim commPort As UInt32
    Dim commBaud As UInt32
    Dim myPKTID As Integer
    Dim registered As Boolean
    Dim ActiveWaveAPI As AW_API_NET.APINetClass = new
AW_API_NET.APINetClass
    Dim ReaderEventHandler As AW_API_NET.fReaderEvent
    Friend WithEvents TextBox1 As System.Windows.Forms.TextBox
    Friend WithEvents TextBox2 As System.Windows.Forms.TextBox
    Friend WithEvents Label13 As System.Windows.Forms.Label
    Friend WithEvents Label14 As System.Windows.Forms.Label
    Friend WithEvents TextBox3 As System.Windows.Forms.TextBox
    Friend WithEvents Label15 As System.Windows.Forms.Label
    Friend WithEvents Timer1 As System.Windows.Forms.Timer
    Dim TagEventHandler As AW_API_NET.fTagEvent
    Dim strCn As String = "Database= dtraxx_db;Data
Source=localhost;User Id=root;Password="
    Friend WithEvents Label16 As System.Windows.Forms.Label
    Friend WithEvents TextBox4 As System.Windows.Forms.TextBox
    Friend WithEvents Label17 As System.Windows.Forms.Label
    Friend WithEvents Label19 As System.Windows.Forms.Label
    Friend WithEvents Button1 As System.Windows.Forms.Button
    Friend WithEvents PictureBox1 As System.Windows.Forms.PictureBox
    Friend WithEvents PictureBox2 As System.Windows.Forms.PictureBox
    Friend WithEvents Button2 As System.Windows.Forms.Button
    Friend WithEvents TextBox6 As System.Windows.Forms.TextBox
    Friend WithEvents Label20 As System.Windows.Forms.Label
```

```
    Friend WithEvents ListView1 As System.Windows.Forms.ListView
    Friend WithEvents ID As System.Windows.Forms.ColumnHeader
    Friend WithEvents Full_Name As System.Windows.Forms.ColumnHeader
    Friend WithEvents Zone As System.Windows.Forms.ColumnHeader
    Friend WithEvents Status As System.Windows.Forms.ColumnHeader
    Friend WithEvents A_Date As System.Windows.Forms.ColumnHeader
    Friend WithEvents A_Time As System.Windows.Forms.ColumnHeader
    Friend WithEvents lblOpName As System.Windows.Forms.Label
    Friend WithEvents commStatus As System.Windows.Forms.Label
    Friend WithEvents lbl_Oname As System.Windows.Forms.Label
    Friend WithEvents lbl_Operator As System.Windows.Forms.Label
    Friend WithEvents BtnOpList As System.Windows.Forms.Button
    Friend WithEvents BtnInmLst As System.Windows.Forms.Button
    Friend WithEvents BtnStart As System.Windows.Forms.Button
    Friend WithEvents BtnAddOfficer As System.Windows.Forms.Button
    Friend WithEvents BtnAddInmate As System.Windows.Forms.Button
    Friend WithEvents BtnLogOut As System.Windows.Forms.Button
    Friend WithEvents TmrCall As System.Windows.Forms.Timer
    Friend WithEvents Button3 As System.Windows.Forms.Button
    Friend WithEvents Label18 As System.Windows.Forms.Label
    Public ofName As String


#Region " Windows Form Designer generated code "

    Public Sub New()
        MyBase.New()

        'This call is required by the Windows Form Designer.
        InitializeComponent()

    End Sub

    Protected Overloads Overrides Sub Dispose(ByVal disposing As
Boolean)
        If disposing Then
            If Not (components Is Nothing) Then
                components.Dispose()
            End If
        End If
        MyBase.Dispose(disposing)
    End Sub

    'Required by the Windows Form Designer
    Private components As System.ComponentModel.IContainer
    Friend WithEvents MsgArea As System.Windows.Forms.TextBox
    Friend WithEvents Label2 As System.Windows.Forms.Label
    Friend WithEvents Button4 As System.Windows.Forms.Button
    Friend WithEvents TagBox As System.Windows.Forms.TextBox
    Friend WithEvents TagButton As System.Windows.Forms.Button
    Friend WithEvents Label3 As System.Windows.Forms.Label
    Friend WithEvents ReaderIDTextBox As System.Windows.Forms.TextBox
    Friend WithEvents ACCRadioButton As
```

```
System.Windows.Forms.RadioButton
    Friend WithEvents ASTRadioButton As
System.Windows.Forms.RadioButton
    Friend WithEvents INVRadioButton As
System.Windows.Forms.RadioButton
    Friend WithEvents QueryButton As System.Windows.Forms.Button
    Friend WithEvents Label4 As System.Windows.Forms.Label
    Friend WithEvents GroupBox1 As System.Windows.Forms.GroupBox
    Friend WithEvents SocketFlg As System.Windows.Forms.CheckBox
    Friend WithEvents GroupBox2 As System.Windows.Forms.GroupBox
    Friend WithEvents GroupBox3 As System.Windows.Forms.GroupBox
    Friend WithEvents Label1 As System.Windows.Forms.Label
    Friend WithEvents Label5 As System.Windows.Forms.Label
    Friend WithEvents rfScanButton As System.Windows.Forms.Button
    Friend WithEvents rfOpenButton As System.Windows.Forms.Button
    Friend WithEvents Label6 As System.Windows.Forms.Label
    Friend WithEvents Label7 As System.Windows.Forms.Label
    Friend WithEvents Label8 As System.Windows.Forms.Label
    Friend WithEvents RdrCmdTypeComboBox As
System.Windows.Forms.ComboBox
    Friend WithEvents TagCmdTypeComboBox As
System.Windows.Forms.ComboBox
    Friend WithEvents Label11 As System.Windows.Forms.Label
    Friend WithEvents rfCloseButton As System.Windows.Forms.Button
    Friend WithEvents rfResetRdrButton As System.Windows.Forms.Button
    Friend WithEvents rfQueryRdrButton As System.Windows.Forms.Button
    Friend WithEvents BroadcastFGenCheckBox As
System.Windows.Forms.CheckBox
    Friend WithEvents rfResetSmartFGenButton As
System.Windows.Forms.Button
    Friend WithEvents rfQuerySTDFGenButton As
System.Windows.Forms.Button
    Friend WithEvents FGenIDTextBox As System.Windows.Forms.TextBox
    Friend WithEvents IPListBox As System.Windows.Forms.ListBox
    Friend WithEvents LengthTextBox As System.Windows.Forms.TextBox
    Friend WithEvents AddressTextBox As System.Windows.Forms.TextBox
    Friend WithEvents ReadTagButton As System.Windows.Forms.Button
    Friend WithEvents EnableTagButton As System.Windows.Forms.Button
    Friend WithEvents ClearButton As System.Windows.Forms.Button
    Friend WithEvents LongIntervalCheckBox As
System.Windows.Forms.CheckBox
    Friend WithEvents Label9 As System.Windows.Forms.Label
    Friend WithEvents STDFGenCheckBox As
System.Windows.Forms.CheckBox
    Friend WithEvents rfSetReaderFSButton As
System.Windows.Forms.Button
    Friend WithEvents rfGetReaderFSButton As
System.Windows.Forms.Button
    Friend WithEvents Label10 As System.Windows.Forms.Label
    Friend WithEvents FSTextBox As System.Windows.Forms.TextBox
    Friend WithEvents Label12 As System.Windows.Forms.Label
    Friend WithEvents GroupBox4 As System.Windows.Forms.GroupBox
    Friend WithEvents GroupBox5 As System.Windows.Forms.GroupBox
    Friend WithEvents GroupBox6 As System.Windows.Forms.GroupBox
```

```
    Friend WithEvents rfCloseSocketButton As
System.Windows.Forms.Button
    Friend WithEvents rfScanIPButton As System.Windows.Forms.Button
    Friend WithEvents SpecificIPRadioButton As
System.Windows.Forms.RadioButton
    Friend WithEvents AllIPRadioButton As
System.Windows.Forms.RadioButton
    Friend WithEvents rfOpenSocketButton As
System.Windows.Forms.Button
    Friend WithEvents IPTextBox As System.Windows.Forms.TextBox
    <System.Diagnostics.DebuggerStepThrough()> Private Sub
InitializeComponent()
        Me.components = New System.ComponentModel.Container
        Dim resources As
System.ComponentModel.ComponentResourceManager = New
System.ComponentModel.ComponentResourceManager(GetType(Form1))
        Me.rfOpenButton = New System.Windows.Forms.Button
        Me.rfCloseButton = New System.Windows.Forms.Button
        Me.MsgArea = New System.Windows.Forms.TextBox
        Me.rfResetRdrButton = New System.Windows.Forms.Button
        Me.TagBox = New System.Windows.Forms.TextBox
        Me.Label2 = New System.Windows.Forms.Label
        Me.TagButton = New System.Windows.Forms.Button
        Me.Label3 = New System.Windows.Forms.Label
        Me.ReaderIDTextBox = New System.Windows.Forms.TextBox
        Me.ACCRadioButton = New System.Windows.Forms.RadioButton
        Me.ASTRadioButton = New System.Windows.Forms.RadioButton
        Me.INVRadioButton = New System.Windows.Forms.RadioButton
        Me.QueryButton = New System.Windows.Forms.Button
        Me.rfQueryRdrButton = New System.Windows.Forms.Button
        Me.TagCmdTypeComboBox = New System.Windows.Forms.ComboBox
        Me.Label4 = New System.Windows.Forms.Label
        Me.GroupBox1 = New System.Windows.Forms.GroupBox
        Me.GroupBox6 = New System.Windows.Forms.GroupBox
        Me.GroupBox5 = New System.Windows.Forms.GroupBox
        Me.IPTextBox = New System.Windows.Forms.TextBox
        Me.AllIPRadioButton = New System.Windows.Forms.RadioButton
        Me.SpecificIPRadioButton = New
System.Windows.Forms.RadioButton
        Me.rfCloseSocketButton = New System.Windows.Forms.Button
        Me.rfScanIPButton = New System.Windows.Forms.Button
        Me.PictureBox2 = New System.Windows.Forms.PictureBox
        Me.PictureBox1 = New System.Windows.Forms.PictureBox
        Me.rfOpenSocketButton = New System.Windows.Forms.Button
        Me.rfScanButton = New System.Windows.Forms.Button
        Me.Label12 = New System.Windows.Forms.Label
        Me.Label10 = New System.Windows.Forms.Label
        Me.FSTextBox = New System.Windows.Forms.TextBox
        Me.rfGetReaderFSButton = New System.Windows.Forms.Button
        Me.rfSetReaderFSButton = New System.Windows.Forms.Button
        Me.SocketFlg = New System.Windows.Forms.CheckBox
        Me.GroupBox2 = New System.Windows.Forms.GroupBox
        Me.Label9 = New System.Windows.Forms.Label
        Me.Label8 = New System.Windows.Forms.Label
        Me.Label7 = New System.Windows.Forms.Label
```

```
Me.LengthTextBox = New System.Windows.Forms.TextBox
Me.AddressTextBox = New System.Windows.Forms.TextBox
Me.ReadTagButton = New System.Windows.Forms.Button
Me.EnableTagButton = New System.Windows.Forms.Button
Me.GroupBox3 = New System.Windows.Forms.GroupBox
Me.STDFGenCheckBox = New System.Windows.Forms.CheckBox
Me.rfQuerySTDFGenButton = New System.Windows.Forms.Button
Me.rfResetSmartFGenButton = New System.Windows.Forms.Button
Me.BroadcastFGenCheckBox = New System.Windows.Forms.CheckBox
Me.Label1 = New System.Windows.Forms.Label
Me.Label5 = New System.Windows.Forms.Label
Me.RdrCmdTypeComboBox = New System.Windows.Forms.ComboBox
Me.Label6 = New System.Windows.Forms.Label
Me.Label11 = New System.Windows.Forms.Label
Me.FGenIDTextBox = New System.Windows.Forms.TextBox
Me.IPListBox = New System.Windows.Forms.ListBox
Me.ClearButton = New System.Windows.Forms.Button
Me.LongIntervalCheckBox = New System.Windows.Forms.CheckBox
Me.GroupBox4 = New System.Windows.Forms.GroupBox
Me.TextBox1 = New System.Windows.Forms.TextBox
Me.TextBox2 = New System.Windows.Forms.TextBox
Me.Label13 = New System.Windows.Forms.Label
Me.Label14 = New System.Windows.Forms.Label
Me.TextBox3 = New System.Windows.Forms.TextBox
Me.Label15 = New System.Windows.Forms.Label
Me.Timer1 = New System.Windows.Forms.Timer(Me.components)
Me.Label16 = New System.Windows.Forms.Label
Me.TextBox4 = New System.Windows.Forms.TextBox
Me.Label17 = New System.Windows.Forms.Label
Me.Label19 = New System.Windows.Forms.Label
Me.Button1 = New System.Windows.Forms.Button
Me.Button2 = New System.Windows.Forms.Button
Me.TextBox6 = New System.Windows.Forms.TextBox
Me.Label20 = New System.Windows.Forms.Label
Me.ListView1 = New System.Windows.Forms.ListView
Me.ID = New System.Windows.Forms.ColumnHeader
Me.Full_Name = New System.Windows.Forms.ColumnHeader
Me.Zone = New System.Windows.Forms.ColumnHeader
Me.Status = New System.Windows.Forms.ColumnHeader
Me.A_Date = New System.Windows.Forms.ColumnHeader
Me.A_Time = New System.Windows.Forms.ColumnHeader
Me.lblOpName = New System.Windows.Forms.Label
Me.commStatus = New System.Windows.Forms.Label
Me.lbl_Oname = New System.Windows.Forms.Label
Me.lbl_Operator = New System.Windows.Forms.Label
Me.BtnOpList = New System.Windows.Forms.Button
Me.BtnInmLst = New System.Windows.Forms.Button
Me.BtnStart = New System.Windows.Forms.Button
Me.BtnAddOfficer = New System.Windows.Forms.Button
Me.BtnAddInmate = New System.Windows.Forms.Button
Me.BtnLogOut = New System.Windows.Forms.Button
Me.TmrCall = New System.Windows.Forms.Timer(Me.components)
Me.Button3 = New System.Windows.Forms.Button
Me.Label18 = New System.Windows.Forms.Label
Me.GroupBox1.SuspendLayout()
```

```
        Me.GroupBox6.SuspendLayout()
        Me.GroupBox5.SuspendLayout()
        CType(Me.PictureBox2,
System.ComponentModel.ISupportInitialize).BeginInit()
        CType(Me.PictureBox1,
System.ComponentModel.ISupportInitialize).BeginInit()
        Me.GroupBox2.SuspendLayout()
        Me.GroupBox3.SuspendLayout()
        Me.GroupBox4.SuspendLayout()
        Me.SuspendLayout()
        '
        'rfOpenButton
        '
        Me.rfOpenButton.Font = New System.Drawing.Font("Microsoft
Sans Serif", 9.0!, System.Drawing.FontStyle.Regular,
System.Drawing.GraphicsUnit.Point, CType(0, Byte))
        Me.rfOpenButton.ForeColor = System.Drawing.Color.Blue
        Me.rfOpenButton.Location = New System.Drawing.Point(16, 26)
        Me.rfOpenButton.Name = "rfOpenButton"
        Me.rfOpenButton.Size = New System.Drawing.Size(120, 26)
        Me.rfOpenButton.TabIndex = 0
        Me.rfOpenButton.Text = "rfOpen"
        '
        'rfCloseButton
        '
        Me.rfCloseButton.Font = New System.Drawing.Font("Microsoft
Sans Serif", 9.0!, System.Drawing.FontStyle.Regular,
System.Drawing.GraphicsUnit.Point, CType(0, Byte))
        Me.rfCloseButton.ForeColor = System.Drawing.Color.Blue
        Me.rfCloseButton.Location = New System.Drawing.Point(16, 58)
        Me.rfCloseButton.Name = "rfCloseButton"
        Me.rfCloseButton.Size = New System.Drawing.Size(120, 26)
        Me.rfCloseButton.TabIndex = 1
        Me.rfCloseButton.Text = "rfClose"
        '
        'MsgArea
        '
        Me.MsgArea.Anchor =
CType((((System.Windows.Forms.AnchorStyles.Top Or
System.Windows.Forms.AnchorStyles.Bottom) _
                    Or System.Windows.Forms.AnchorStyles.Left) _
                    Or System.Windows.Forms.AnchorStyles.Right),
System.Windows.Forms.AnchorStyles)
        Me.MsgArea.Font = New System.Drawing.Font("Microsoft Sans
Serif", 8.25!, System.Drawing.FontStyle.Regular,
System.Drawing.GraphicsUnit.Point, CType(0, Byte))
        Me.MsgArea.Location = New System.Drawing.Point(123, 549)
        Me.MsgArea.Multiline = True
        Me.MsgArea.Name = "MsgArea"
        Me.MsgArea.ScrollBars = System.Windows.Forms.ScrollBars.Both
        Me.MsgArea.Size = New System.Drawing.Size(309, 45)
        Me.MsgArea.TabIndex = 3
        Me.MsgArea.Visible = False
        '
        'rfResetRdrButton
```

```
        Me.rfResetRdrButton.Font = New System.Drawing.Font("Microsoft
Sans Serif", 9.0!, System.Drawing.FontStyle.Regular,
System.Drawing.GraphicsUnit.Point, CType(0, Byte))
        Me.rfResetRdrButton.ForeColor = System.Drawing.Color.Blue
        Me.rfResetRdrButton.Location = New System.Drawing.Point(12,
38)
        Me.rfResetRdrButton.Name = "rfResetRdrButton"
        Me.rfResetRdrButton.Size = New System.Drawing.Size(96, 26)
        Me.rfResetRdrButton.TabIndex = 5
        Me.rfResetRdrButton.Text = "rfResetReader"

        Me.TagBox.Font = New System.Drawing.Font("Microsoft Sans
Serif", 9.0!, System.Drawing.FontStyle.Regular,
System.Drawing.GraphicsUnit.Point, CType(0, Byte))
        Me.TagBox.Location = New System.Drawing.Point(615, 554)
        Me.TagBox.Name = "TagBox"
        Me.TagBox.Size = New System.Drawing.Size(90, 21)
        Me.TagBox.TabIndex = 6
        Me.TagBox.Visible = False
                '
        Me.Label2.Font = New System.Drawing.Font("Microsoft Sans
Serif", 9.0!, System.Drawing.FontStyle.Regular,
System.Drawing.GraphicsUnit.Point, CType(0, Byte))
        Me.Label2.Location = New System.Drawing.Point(565, 556)
        Me.Label2.Name = "Label2"
        Me.Label2.Size = New System.Drawing.Size(48, 18)
        Me.Label2.TabIndex = 7
        Me.Label2.Text = "Tag ID: "
        Me.Label2.Visible = False

        Me.TagButton.Font = New System.Drawing.Font("Microsoft Sans
Serif", 9.0!, System.Drawing.FontStyle.Regular,
System.Drawing.GraphicsUnit.Point, CType(0, Byte))
        Me.TagButton.ForeColor = System.Drawing.Color.Blue
        Me.TagButton.Location = New System.Drawing.Point(16, 55)
        Me.TagButton.Name = "TagButton"
        Me.TagButton.Size = New System.Drawing.Size(144, 23)
        Me.TagButton.TabIndex = 8
        Me.TagButton.Text = "rfCallTag"
        Me.TagButton.Visible = False

        Me.Label3.Font = New System.Drawing.Font("Microsoft Sans
Serif", 9.0!, System.Drawing.FontStyle.Regular,
System.Drawing.GraphicsUnit.Point, CType(0, Byte))
        Me.Label3.Location = New System.Drawing.Point(387, 554)
        Me.Label3.Name = "Label3"
        Me.Label3.Size = New System.Drawing.Size(64, 18)
        Me.Label3.TabIndex = 9
        Me.Label3.Text = "Reader ID:"
        Me.Label3.Visible = False
        Me.ReaderIDTextBox.Font = New System.Drawing.Font("Microsoft
Sans Serif", 9.0!, System.Drawing.FontStyle.Regular,
System.Drawing.GraphicsUnit.Point, CType(0, Byte))
        Me.ReaderIDTextBox.Location = New System.Drawing.Point(453,
552)
```

```
        Me.ReaderIDTextBox.Name = "ReaderIDTextBox"
        Me.ReaderIDTextBox.Size = New System.Drawing.Size(54, 21)
        Me.ReaderIDTextBox.TabIndex = 10
        Me.ReaderIDTextBox.Visible = False
        Me.ACCRadioButton.Checked = True
        Me.ACCRadioButton.Font = New System.Drawing.Font("Microsoft
Sans Serif", 9.0!, System.Drawing.FontStyle.Regular,
System.Drawing.GraphicsUnit.Point, CType(0, Byte))
        Me.ACCRadioButton.Location = New System.Drawing.Point(727,
556)
        Me.ACCRadioButton.Name = "ACCRadioButton"
        Me.ACCRadioButton.Size = New System.Drawing.Size(64, 24)
        Me.ACCRadioButton.TabIndex = 11
        Me.ACCRadioButton.TabStop = True
        Me.ACCRadioButton.Text = "Access"
        Me.ACCRadioButton.Visible = False
        Me.ASTRadioButton.Font = New System.Drawing.Font("Microsoft
Sans Serif", 9.0!, System.Drawing.FontStyle.Regular,
System.Drawing.GraphicsUnit.Point, CType(0, Byte))
        Me.ASTRadioButton.Location = New System.Drawing.Point(727,
582)
        Me.ASTRadioButton.Name = "ASTRadioButton"
        Me.ASTRadioButton.Size = New System.Drawing.Size(64, 24)
        Me.ASTRadioButton.TabIndex = 12
        Me.ASTRadioButton.Text = "Asset"
        Me.ASTRadioButton.Visible = False
        Me.INVRadioButton.Font = New System.Drawing.Font("Microsoft
Sans Serif", 9.0!, System.Drawing.FontStyle.Regular,
System.Drawing.GraphicsUnit.Point, CType(0, Byte))
        Me.INVRadioButton.Location = New System.Drawing.Point(727,
608)
        Me.INVRadioButton.Name = "INVRadioButton"
        Me.INVRadioButton.Size = New System.Drawing.Size(76, 24)
        Me.INVRadioButton.TabIndex = 13
        Me.INVRadioButton.Text = "Inventory"
        Me.INVRadioButton.Visible = False
        Me.QueryButton.BackColor =
System.Drawing.Color.FromArgb(CType(CType(255, Byte), Integer),
CType(CType(192, Byte), Integer), CType(CType(255, Byte), Integer))
        Me.QueryButton.Font = New System.Drawing.Font("Franklin
Gothic Book", 9.75!, System.Drawing.FontStyle.Bold,
System.Drawing.GraphicsUnit.Point, CType(0, Byte))
        Me.QueryButton.ForeColor = System.Drawing.Color.Crimson
        Me.QueryButton.Location = New System.Drawing.Point(229, 436)
        Me.QueryButton.Name = "QueryButton"
        Me.QueryButton.Size = New System.Drawing.Size(120, 42)
        Me.QueryButton.TabIndex = 14
        Me.QueryButton.Text = "Calling Tag"
        Me.QueryButton.UseVisualStyleBackColor = False
        Me.rfQueryRdrButton.Font = New System.Drawing.Font("Microsoft
Sans Serif", 9.0!, System.Drawing.FontStyle.Regular,
System.Drawing.GraphicsUnit.Point, CType(0, Byte))
        Me.rfQueryRdrButton.ForeColor = System.Drawing.Color.Blue
        Me.rfQueryRdrButton.Location = New System.Drawing.Point(12,
68)
```

```
        Me.rfQueryRdrButton.Name = "rfQueryRdrButton"
        Me.rfQueryRdrButton.Size = New System.Drawing.Size(96, 26)
        Me.rfQueryRdrButton.TabIndex = 16
        Me.rfQueryRdrButton.Text = "rfQuryReader"
        Me.TagCmdTypeComboBox.Font = New
System.Drawing.Font("Microsoft Sans Serif", 9.0!,
System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point,
CType(0, Byte))
        Me.TagCmdTypeComboBox.Items.AddRange(New Object()
{"RF_SELECT_TAG_ID", "RF_SELECT_FIELD", "RF_SELECT_TAG_TYPE",
"RF_SELECT_TAG_RANGE"})
        Me.TagCmdTypeComboBox.Location = New
System.Drawing.Point(563, 606)
        Me.TagCmdTypeComboBox.Name = "TagCmdTypeComboBox"
        Me.TagCmdTypeComboBox.Size = New System.Drawing.Size(160, 23)
        Me.TagCmdTypeComboBox.TabIndex = 17
        Me.TagCmdTypeComboBox.Text = "RF_SELECT_TAG_TYPE"
        Me.TagCmdTypeComboBox.Visible = False
        Me.Label4.Font = New System.Drawing.Font("Microsoft Sans
Serif", 9.0!, System.Drawing.FontStyle.Regular,
System.Drawing.GraphicsUnit.Point, CType(0, Byte))
        Me.Label4.Location = New System.Drawing.Point(387, 584)
        Me.Label4.Name = "Label4"
        Me.Label4.Size = New System.Drawing.Size(120, 16)
        Me.Label4.TabIndex = 18
        Me.Label4.Text = "Reader Cmd Type : "
        Me.Label4.Visible = False
        Me.GroupBox1.Controls.Add(Me.GroupBox6)
        Me.GroupBox1.Controls.Add(Me.GroupBox5)
        Me.GroupBox1.Font = New System.Drawing.Font("Microsoft Sans
Serif", 9.0!, System.Drawing.FontStyle.Regular,
System.Drawing.GraphicsUnit.Point, CType(0, Byte))
        Me.GroupBox1.Location = New System.Drawing.Point(390, 550)
        Me.GroupBox1.Name = "GroupBox1"
        Me.GroupBox1.Size = New System.Drawing.Size(194, 104)
        Me.GroupBox1.TabIndex = 19
        Me.GroupBox1.TabStop = False
        Me.GroupBox1.Text = "Communication"
        Me.GroupBox1.Visible = False
        Me.GroupBox6.Controls.Add(Me.rfOpenButton)
        Me.GroupBox6.Controls.Add(Me.rfCloseButton)
        Me.GroupBox6.Location = New System.Drawing.Point(16, 248)
        Me.GroupBox6.Name = "GroupBox6"
        Me.GroupBox6.Size = New System.Drawing.Size(148, 98)
        Me.GroupBox6.TabIndex = 22
        Me.GroupBox6.TabStop = False
        Me.GroupBox6.Text = "RS-232"
        Me.GroupBox6.Visible = False
        Me.GroupBox5.Controls.Add(Me.IPTextBox)
        Me.GroupBox5.Controls.Add(Me.AllIPRadioButton)
        Me.GroupBox5.Controls.Add(Me.SpecificIPRadioButton)
        Me.GroupBox5.Controls.Add(Me.rfCloseSocketButton)
        Me.GroupBox5.Controls.Add(Me.rfScanIPButton)
        Me.GroupBox5.Location = New System.Drawing.Point(16, 20)
        Me.GroupBox5.Name = "GroupBox5"
```

```
        Me.GroupBox5.Size = New System.Drawing.Size(172, 78)
        Me.GroupBox5.TabIndex = 21
        Me.GroupBox5.TabStop = False
        Me.GroupBox5.Text = "Network"
        Me.IPTextBox.Font = New System.Drawing.Font("Microsoft Sans
Serif", 9.0!, System.Drawing.FontStyle.Regular,
System.Drawing.GraphicsUnit.Point, CType(0, Byte))
        Me.IPTextBox.Location = New System.Drawing.Point(20, 110)
        Me.IPTextBox.Name = "IPTextBox"
        Me.IPTextBox.ReadOnly = True
        Me.IPTextBox.Size = New System.Drawing.Size(118, 21)
        Me.IPTextBox.TabIndex = 23
        Me.IPTextBox.Visible = False
        Me.AllIPRadioButton.Checked = True
        Me.AllIPRadioButton.Location = New System.Drawing.Point(92,
144)
        Me.AllIPRadioButton.Name = "AllIPRadioButton"
        Me.AllIPRadioButton.Size = New System.Drawing.Size(52, 24)
        Me.AllIPRadioButton.TabIndex = 22
        Me.AllIPRadioButton.TabStop = True
        Me.AllIPRadioButton.Text = "All IP"
        Me.AllIPRadioButton.Visible = False
        Me.SpecificIPRadioButton.Location = New
System.Drawing.Point(3, 142)
        Me.SpecificIPRadioButton.Name = "SpecificIPRadioButton"
        Me.SpecificIPRadioButton.Size = New System.Drawing.Size(84,
24)
        Me.SpecificIPRadioButton.TabIndex = 21
        Me.SpecificIPRadioButton.Text = "Specific IP"
        Me.SpecificIPRadioButton.Visible = False
        Me.rfCloseSocketButton.Font = New
System.Drawing.Font("Microsoft Sans Serif", 9.0!,
System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point,
CType(0, Byte))
        Me.rfCloseSocketButton.ForeColor = System.Drawing.Color.Blue
        Me.rfCloseSocketButton.Location = New
System.Drawing.Point(18, 174)
        Me.rfCloseSocketButton.Name = "rfCloseSocketButton"
        Me.rfCloseSocketButton.Size = New System.Drawing.Size(120,
26)
        Me.rfCloseSocketButton.TabIndex = 20
        Me.rfCloseSocketButton.Text = "rfCloseSocket"
        Me.rfCloseSocketButton.Visible = False
        Me.rfScanIPButton.Enabled = False
        Me.rfScanIPButton.Font = New System.Drawing.Font("Microsoft
Sans Serif", 9.0!, System.Drawing.FontStyle.Regular,
System.Drawing.GraphicsUnit.Point, CType(0, Byte))
        Me.rfScanIPButton.ForeColor = System.Drawing.Color.Blue
        Me.rfScanIPButton.Location = New System.Drawing.Point(20, 82)
        Me.rfScanIPButton.Name = "rfScanIPButton"
        Me.rfScanIPButton.Size = New System.Drawing.Size(120, 26)
        Me.rfScanIPButton.TabIndex = 19
        Me.rfScanIPButton.Text = "rfScanIP"
        Me.rfScanIPButton.Visible = False
```

```
        Me.PictureBox2.Image =
CType(resources.GetObject("PictureBox2.Image"), System.Drawing.Image)
        Me.PictureBox2.Location = New System.Drawing.Point(192, 445)
        Me.PictureBox2.Name = "PictureBox2"
        Me.PictureBox2.Size = New System.Drawing.Size(22, 23)
        Me.PictureBox2.SizeMode =
System.Windows.Forms.PictureBoxSizeMode.StretchImage
        Me.PictureBox2.TabIndex = 45
        Me.PictureBox2.TabStop = False
        Me.PictureBox2.Visible = False
        Me.PictureBox1.ErrorImage = Nothing
        Me.PictureBox1.Image =
CType(resources.GetObject("PictureBox1.Image"), System.Drawing.Image)
        Me.PictureBox1.InitialImage =
CType(resources.GetObject("PictureBox1.InitialImage"),
System.Drawing.Image)
        Me.PictureBox1.Location = New System.Drawing.Point(192, 394)
        Me.PictureBox1.Name = "PictureBox1"
        Me.PictureBox1.Size = New System.Drawing.Size(22, 24)
        Me.PictureBox1.SizeMode =
System.Windows.Forms.PictureBoxSizeMode.StretchImage
        Me.PictureBox1.TabIndex = 45
        Me.PictureBox1.TabStop = False
        Me.PictureBox1.Visible = False
        Me.rfOpenSocketButton.BackColor =
System.Drawing.Color.FromArgb(CType(CType(255, Byte), Integer),
CType(CType(192, Byte), Integer), CType(CType(255, Byte), Integer))
        Me.rfOpenSocketButton.Font = New
System.Drawing.Font("Franklin Gothic Book", 9.75!,
System.Drawing.FontStyle.Bold, System.Drawing.GraphicsUnit.Point,
CType(0, Byte))
        Me.rfOpenSocketButton.ForeColor =
System.Drawing.Color.Crimson
        Me.rfOpenSocketButton.Location = New System.Drawing.Point(71,
436)
        Me.rfOpenSocketButton.Name = "rfOpenSocketButton"
        Me.rfOpenSocketButton.Size = New System.Drawing.Size(120, 42)
        Me.rfOpenSocketButton.TabIndex = 24
        Me.rfOpenSocketButton.Text = "Open Socket"
        Me.rfOpenSocketButton.UseVisualStyleBackColor = False
        Me.rfScanButton.BackColor =
System.Drawing.Color.FromArgb(CType(CType(255, Byte), Integer),
CType(CType(192, Byte), Integer), CType(CType(255, Byte), Integer))
        Me.rfScanButton.Font = New System.Drawing.Font("Franklin
Gothic Book", 9.75!, System.Drawing.FontStyle.Bold,
System.Drawing.GraphicsUnit.Point, CType(0, Byte))
        Me.rfScanButton.ForeColor = System.Drawing.Color.Crimson
        Me.rfScanButton.Location = New System.Drawing.Point(71, 381)
        Me.rfScanButton.Name = "rfScanButton"
        Me.rfScanButton.Size = New System.Drawing.Size(120, 49)
        Me.rfScanButton.TabIndex = 18
        Me.rfScanButton.Text = "Scan"
        Me.rfScanButton.UseVisualStyleBackColor = False
        Me.Label12.Anchor = System.Windows.Forms.AnchorStyles.Top
```

```
        Me.Label12.Font = New System.Drawing.Font("Microsoft Sans
Serif", 8.25!, System.Drawing.FontStyle.Regular,
System.Drawing.GraphicsUnit.Point, CType(0, Byte))
        Me.Label12.Location = New System.Drawing.Point(172, 76)
        Me.Label12.Name = "Label12"
        Me.Label12.Size = New System.Drawing.Size(76, 12)
        Me.Label12.TabIndex = 23
        Me.Label12.Text = "FS Limit (0 - 20)"
        Me.Label10.Font = New System.Drawing.Font("Microsoft Sans
Serif", 9.0!, System.Drawing.FontStyle.Regular,
System.Drawing.GraphicsUnit.Point, CType(0, Byte))
        Me.Label10.Location = New System.Drawing.Point(136, 54)
        Me.Label10.Name = "Label10"
        Me.Label10.Size = New System.Drawing.Size(32, 16)
        Me.Label10.TabIndex = 22
        Me.Label10.Text = "FS : "
        Me.Label10.TextAlign =
System.Drawing.ContentAlignment.MiddleRight
        Me.FSTextBox.Font = New System.Drawing.Font("Microsoft Sans
Serif", 9.0!, System.Drawing.FontStyle.Regular,
System.Drawing.GraphicsUnit.Point, CType(0, Byte))
        Me.FSTextBox.Location = New System.Drawing.Point(172, 52)
        Me.FSTextBox.Name = "FSTextBox"
        Me.FSTextBox.Size = New System.Drawing.Size(68, 21)
        Me.FSTextBox.TabIndex = 21
        Me.rfGetReaderFSButton.Font = New
System.Drawing.Font("Microsoft Sans Serif", 9.0!,
System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point,
CType(0, Byte))
        Me.rfGetReaderFSButton.ForeColor = System.Drawing.Color.Blue
        Me.rfGetReaderFSButton.Location = New
System.Drawing.Point(136, 92)
        Me.rfGetReaderFSButton.Name = "rfGetReaderFSButton"
        Me.rfGetReaderFSButton.Size = New System.Drawing.Size(106,
23)
        Me.rfGetReaderFSButton.TabIndex = 20
        Me.rfGetReaderFSButton.Text = "rfGetReaderFS"
        Me.rfSetReaderFSButton.Font = New
System.Drawing.Font("Microsoft Sans Serif", 9.0!,
System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point,
CType(0, Byte))
        Me.rfSetReaderFSButton.ForeColor = System.Drawing.Color.Blue
        Me.rfSetReaderFSButton.Location = New
System.Drawing.Point(142, 26)
        Me.rfSetReaderFSButton.Name = "rfSetReaderFSButton"
        Me.rfSetReaderFSButton.Size = New System.Drawing.Size(98, 23)
        Me.rfSetReaderFSButton.TabIndex = 19
        Me.rfSetReaderFSButton.Text = "rfSetReaderFS"
        Me.SocketFlg.Font = New System.Drawing.Font("Microsoft Sans
Serif", 9.0!, System.Drawing.FontStyle.Regular,
System.Drawing.GraphicsUnit.Point, CType(0, Byte))
        Me.SocketFlg.Location = New System.Drawing.Point(361, 596)
        Me.SocketFlg.Name = "SocketFlg"
        Me.SocketFlg.Size = New System.Drawing.Size(16, 14)
        Me.SocketFlg.TabIndex = 17
```

```
        Me.SocketFlg.Text = "Open Using Socket"
        Me.SocketFlg.Visible = False
        Me.GroupBox2.Controls.Add(Me.Label9)
        Me.GroupBox2.Controls.Add(Me.Label8)
        Me.GroupBox2.Controls.Add(Me.Label7)
        Me.GroupBox2.Controls.Add(Me.LengthTextBox)
        Me.GroupBox2.Controls.Add(Me.AddressTextBox)
        Me.GroupBox2.Controls.Add(Me.ReadTagButton)
        Me.GroupBox2.Controls.Add(Me.EnableTagButton)
        Me.GroupBox2.Controls.Add(Me.TagButton)
        Me.GroupBox2.Font = New System.Drawing.Font("Microsoft Sans
Serif", 9.0!, System.Drawing.FontStyle.Regular,
System.Drawing.GraphicsUnit.Point, CType(0, Byte))
        Me.GroupBox2.Location = New System.Drawing.Point(390, 706)
        Me.GroupBox2.Name = "GroupBox2"
        Me.GroupBox2.Size = New System.Drawing.Size(160, 53)
        Me.GroupBox2.TabIndex = 20
        Me.GroupBox2.TabStop = False
        Me.GroupBox2.Text = "Tag"
        Me.GroupBox2.Visible = False
        Me.Label9.Font = New System.Drawing.Font("Microsoft Sans
Serif", 9.0!, System.Drawing.FontStyle.Regular,
System.Drawing.GraphicsUnit.Point, CType(0, Byte))
        Me.Label9.Location = New System.Drawing.Point(162, 152)
        Me.Label9.Name = "Label9"
        Me.Label9.Size = New System.Drawing.Size(38, 16)
        Me.Label9.TabIndex = 21
        Me.Label9.Text = "(Hex)"
        Me.Label9.Visible = False
        Me.Label8.Font = New System.Drawing.Font("Microsoft Sans
Serif", 9.0!, System.Drawing.FontStyle.Regular,
System.Drawing.GraphicsUnit.Point, CType(0, Byte))
        Me.Label8.Location = New System.Drawing.Point(58, 178)
        Me.Label8.Name = "Label8"
        Me.Label8.Size = New System.Drawing.Size(52, 16)
        Me.Label8.TabIndex = 20
        Me.Label8.Text = "Length : "
        Me.Label8.Visible = False
        Me.Label7.Font = New System.Drawing.Font("Microsoft Sans
Serif", 9.0!, System.Drawing.FontStyle.Regular,
System.Drawing.GraphicsUnit.Point, CType(0, Byte))
        Me.Label7.Location = New System.Drawing.Point(50, 152)
        Me.Label7.Name = "Label7"
        Me.Label7.Size = New System.Drawing.Size(60, 16)
        Me.Label7.TabIndex = 19
        Me.Label7.Text = "Address : "
        Me.Label7.Visible = False
        Me.LengthTextBox.Font = New System.Drawing.Font("Microsoft
Sans Serif", 9.0!, System.Drawing.FontStyle.Regular,
System.Drawing.GraphicsUnit.Point, CType(0, Byte))
       Me.LengthTextBox.Location = New System.Drawing.Point(110, 174)
        Me.LengthTextBox.Name = "LengthTextBox"
        Me.LengthTextBox.Size = New System.Drawing.Size(50, 21)
        Me.LengthTextBox.TabIndex = 18
        Me.LengthTextBox.Visible = False
```

```
        Me.AddressTextBox.Font = New System.Drawing.Font("Microsoft
Sans Serif", 9.0!, System.Drawing.FontStyle.Regular,
System.Drawing.GraphicsUnit.Point, CType(0, Byte))
        Me.AddressTextBox.Location = New System.Drawing.Point(110,
150)
        Me.AddressTextBox.Name = "AddressTextBox"
        Me.AddressTextBox.Size = New System.Drawing.Size(50, 21)
        Me.AddressTextBox.TabIndex = 17
        Me.AddressTextBox.Text = "E0"
        Me.AddressTextBox.Visible = False
        Me.ReadTagButton.Font = New System.Drawing.Font("Microsoft
Sans Serif", 9.0!, System.Drawing.FontStyle.Regular,
System.Drawing.GraphicsUnit.Point, CType(0, Byte))
        Me.ReadTagButton.ForeColor = System.Drawing.Color.Blue
        Me.ReadTagButton.Location = New System.Drawing.Point(50, 124)
        Me.ReadTagButton.Name = "ReadTagButton"
        Me.ReadTagButton.Size = New System.Drawing.Size(144, 23)
        Me.ReadTagButton.TabIndex = 16
        Me.ReadTagButton.Text = "rfReadTag"
        Me.ReadTagButton.Visible = False
        Me.EnableTagButton.Font = New System.Drawing.Font("Microsoft
Sans Serif", 9.0!, System.Drawing.FontStyle.Regular,
System.Drawing.GraphicsUnit.Point, CType(0, Byte))
        Me.EnableTagButton.ForeColor = System.Drawing.Color.Blue
        Me.EnableTagButton.Location = New System.Drawing.Point(50,
84)
        Me.EnableTagButton.Name = "EnableTagButton"
        Me.EnableTagButton.Size = New System.Drawing.Size(144, 23)
        Me.EnableTagButton.TabIndex = 15
        Me.EnableTagButton.Text = "rfEnableTag"
        Me.EnableTagButton.Visible = False
        Me.GroupBox3.Controls.Add(Me.STDFGenCheckBox)
        Me.GroupBox3.Controls.Add(Me.rfQuerySTDFGenButton)
        Me.GroupBox3.Controls.Add(Me.rfResetSmartFGenButton)
        Me.GroupBox3.Controls.Add(Me.BroadcastFGenCheckBox)
        Me.GroupBox3.Font = New System.Drawing.Font("Microsoft Sans
Serif", 9.0!, System.Drawing.FontStyle.Regular,
System.Drawing.GraphicsUnit.Point, CType(0, Byte))
        Me.GroupBox3.Location = New System.Drawing.Point(707, 638)
        Me.GroupBox3.Name = "GroupBox3"
        Me.GroupBox3.Size = New System.Drawing.Size(200, 184)
        Me.GroupBox3.TabIndex = 21
        Me.GroupBox3.TabStop = False
        Me.GroupBox3.Text = "Field Generator"
        Me.GroupBox3.Visible = False
        Me.STDFGenCheckBox.Font = New System.Drawing.Font("Microsoft
Sans Serif", 9.0!, System.Drawing.FontStyle.Regular,
System.Drawing.GraphicsUnit.Point, CType(0, Byte))
        Me.STDFGenCheckBox.Location = New System.Drawing.Point(16,
54)
        Me.STDFGenCheckBox.Name = "STDFGenCheckBox"
        Me.STDFGenCheckBox.Size = New System.Drawing.Size(176, 24)
        Me.STDFGenCheckBox.TabIndex = 26
        Me.STDFGenCheckBox.Text = "STD Field Gen (9600 baud)"
```

```
        Me.rfQuerySTDFGenButton.Font = New
System.Drawing.Font("Microsoft Sans Serif", 9.0!,
System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point,
CType(0, Byte))
        Me.rfQuerySTDFGenButton.ForeColor = System.Drawing.Color.Blue
        Me.rfQuerySTDFGenButton.Location = New
System.Drawing.Point(32, 30)
        Me.rfQuerySTDFGenButton.Name = "rfQuerySTDFGenButton"
        Me.rfQuerySTDFGenButton.Size = New System.Drawing.Size(144,
23)
        Me.rfQuerySTDFGenButton.TabIndex = 20
        Me.rfQuerySTDFGenButton.Text = "rfQuerySTDFGen"
        Me.rfResetSmartFGenButton.Font = New
System.Drawing.Font("Microsoft Sans Serif", 9.0!,
System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point,
CType(0, Byte))
        Me.rfResetSmartFGenButton.ForeColor =
System.Drawing.Color.Blue
        Me.rfResetSmartFGenButton.Location = New
System.Drawing.Point(34, 124)
        Me.rfResetSmartFGenButton.Name = "rfResetSmartFGenButton"
        Me.rfResetSmartFGenButton.Size = New System.Drawing.Size(144,
23)
        Me.rfResetSmartFGenButton.TabIndex = 22
        Me.rfResetSmartFGenButton.Text = "rfResetSmartFGen"
        Me.BroadcastFGenCheckBox.Font = New
System.Drawing.Font("Microsoft Sans Serif", 9.0!,
System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point,
CType(0, Byte))
        Me.BroadcastFGenCheckBox.Location = New
System.Drawing.Point(20, 144)
        Me.BroadcastFGenCheckBox.Name = "BroadcastFGenCheckBox"
        Me.BroadcastFGenCheckBox.Size = New System.Drawing.Size(84,
24)
        Me.BroadcastFGenCheckBox.TabIndex = 25
        Me.BroadcastFGenCheckBox.Text = "Broadcast"

        Me.Label1.Font = New System.Drawing.Font("Microsoft Sans
Serif", 9.0!, System.Drawing.FontStyle.Regular,
System.Drawing.GraphicsUnit.Point, CType(0, Byte))
        Me.Label1.Location = New System.Drawing.Point(367, 678)
        Me.Label1.Name = "Label1"
        Me.Label1.Size = New System.Drawing.Size(88, 16)
        Me.Label1.TabIndex = 22
        Me.Label1.Text = "Message List :"
        Me.Label1.Visible = False
        Me.Label5.Font = New System.Drawing.Font("Microsoft Sans
Serif", 9.0!, System.Drawing.FontStyle.Regular,
System.Drawing.GraphicsUnit.Point, CType(0, Byte))
        Me.Label5.Location = New System.Drawing.Point(313, 556)
        Me.Label5.Name = "Label5"
        Me.Label5.Size = New System.Drawing.Size(96, 16)
        Me.Label5.TabIndex = 24
        Me.Label5.Text = "IP Address List :"
        Me.Label5.Visible = False
```

```
        Me.RdrCmdTypeComboBox.Font = New
System.Drawing.Font("Microsoft Sans Serif", 9.0!,
System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point,
CType(0, Byte))
        Me.RdrCmdTypeComboBox.Items.AddRange(New Object()
{"SPECIFIC_READER", "ALL_READERS"})
        Me.RdrCmdTypeComboBox.Location = New
System.Drawing.Point(385, 602)
        Me.RdrCmdTypeComboBox.Name = "RdrCmdTypeComboBox"
        Me.RdrCmdTypeComboBox.Size = New System.Drawing.Size(156, 23)
        Me.RdrCmdTypeComboBox.TabIndex = 25
        Me.RdrCmdTypeComboBox.Text = "ALL_READERS"
        Me.RdrCmdTypeComboBox.Visible = False
        Me.Label6.Font = New System.Drawing.Font("Microsoft Sans
Serif", 9.0!, System.Drawing.FontStyle.Regular,
System.Drawing.GraphicsUnit.Point, CType(0, Byte))
        Me.Label6.Location = New System.Drawing.Point(565, 588)
        Me.Label6.Name = "Label6"
        Me.Label6.Size = New System.Drawing.Size(120, 16)
        Me.Label6.TabIndex = 26
        Me.Label6.Text = "Tag Cmd Type : "
        Me.Label6.Visible = False
        Me.Label11.Font = New System.Drawing.Font("Microsoft Sans
Serif", 9.0!, System.Drawing.FontStyle.Regular,
System.Drawing.GraphicsUnit.Point, CType(0, Byte))
        Me.Label11.Location = New System.Drawing.Point(711, 554)
        Me.Label11.Name = "Label11"
        Me.Label11.Size = New System.Drawing.Size(56, 18)
        Me.Label11.TabIndex = 28
        Me.Label11.Text = "FGen ID: "
        Me.Label11.Visible = False
        Me.FGenIDTextBox.Font = New System.Drawing.Font("Microsoft
Sans Serif", 9.0!, System.Drawing.FontStyle.Regular,
System.Drawing.GraphicsUnit.Point, CType(0, Byte))
        Me.FGenIDTextBox.Location = New System.Drawing.Point(769,
552)
        Me.FGenIDTextBox.Name = "FGenIDTextBox"
        Me.FGenIDTextBox.Size = New System.Drawing.Size(90, 21)
        Me.FGenIDTextBox.TabIndex = 27
        Me.FGenIDTextBox.Visible = False
        Me.IPListBox.Font = New System.Drawing.Font("Microsoft Sans
Serif", 9.0!, System.Drawing.FontStyle.Regular,
System.Drawing.GraphicsUnit.Point, CType(0, Byte))
        Me.IPListBox.ItemHeight = 15
        Me.IPListBox.Location = New System.Drawing.Point(230, 381)
        Me.IPListBox.Name = "IPListBox"
        Me.IPListBox.Size = New System.Drawing.Size(119, 49)
        Me.IPListBox.TabIndex = 29
        Me.ClearButton.Font = New System.Drawing.Font("Microsoft Sans
Serif", 9.0!, System.Drawing.FontStyle.Regular,
System.Drawing.GraphicsUnit.Point, CType(0, Byte))
        Me.ClearButton.ForeColor = System.Drawing.Color.Blue
        Me.ClearButton.Location = New System.Drawing.Point(772, 736)
        Me.ClearButton.Name = "ClearButton"
        Me.ClearButton.Size = New System.Drawing.Size(50, 26)
```

```
        Me.ClearButton.TabIndex = 30
        Me.ClearButton.Text = "Clear"
        Me.ClearButton.Visible = False

        Me.LongIntervalCheckBox.Font = New
System.Drawing.Font("Microsoft Sans Serif", 9.0!,
System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point,
CType(0, Byte))
        Me.LongIntervalCheckBox.Location = New
System.Drawing.Point(711, 578)
        Me.LongIntervalCheckBox.Name = "LongIntervalCheckBox"
        Me.LongIntervalCheckBox.Size = New System.Drawing.Size(100,
24)
        Me.LongIntervalCheckBox.TabIndex = 31
        Me.LongIntervalCheckBox.Text = "Long Interval"
        Me.LongIntervalCheckBox.Visible = False
        Me.GroupBox4.Controls.Add(Me.rfResetRdrButton)
        Me.GroupBox4.Controls.Add(Me.rfQueryRdrButton)
        Me.GroupBox4.Controls.Add(Me.Label12)
        Me.GroupBox4.Controls.Add(Me.Label10)
        Me.GroupBox4.Controls.Add(Me.FSTextBox)
        Me.GroupBox4.Controls.Add(Me.rfGetReaderFSButton)
        Me.GroupBox4.Controls.Add(Me.rfSetReaderFSButton)
        Me.GroupBox4.Font = New System.Drawing.Font("Microsoft Sans
Serif", 9.0!, System.Drawing.FontStyle.Regular,
System.Drawing.GraphicsUnit.Point, CType(0, Byte))
        Me.GroupBox4.Location = New System.Drawing.Point(304, 584)
        Me.GroupBox4.Name = "GroupBox4"
        Me.GroupBox4.Size = New System.Drawing.Size(260, 134)
        Me.GroupBox4.TabIndex = 32
        Me.GroupBox4.TabStop = False
        Me.GroupBox4.Text = "Reader"
        Me.GroupBox4.Visible = False
        Me.TextBox1.Location = New System.Drawing.Point(684, 550)
        Me.TextBox1.Name = "TextBox1"
        Me.TextBox1.Size = New System.Drawing.Size(66, 20)
        Me.TextBox1.TabIndex = 33
        Me.TextBox1.Visible = False
        Me.TextBox2.Location = New System.Drawing.Point(684, 574)
        Me.TextBox2.Name = "TextBox2"
        Me.TextBox2.Size = New System.Drawing.Size(66, 20)
        Me.TextBox2.TabIndex = 34
        Me.TextBox2.Visible = False
        Me.Label13.AutoSize = True
        Me.Label13.Location = New System.Drawing.Point(606, 557)
        Me.Label13.Name = "Label13"
        Me.Label13.Size = New System.Drawing.Size(47, 13)
        Me.Label13.TabIndex = 35
        Me.Label13.Text = "Room :"
        Me.Label13.Visible = False
        Me.Label14.AutoSize = True
        Me.Label14.Location = New System.Drawing.Point(606, 581)
        Me.Label14.Name = "Label14"
        Me.Label14.Size = New System.Drawing.Size(50, 13)
        Me.Label14.TabIndex = 36
```

```
        Me.Label14.Text = "Tag ID:"
        Me.Label14.Visible = False
        Me.TextBox3.Location = New System.Drawing.Point(684, 626)
        Me.TextBox3.Name = "TextBox3"
        Me.TextBox3.Size = New System.Drawing.Size(66, 20)
        Me.TextBox3.TabIndex = 37
        Me.TextBox3.Visible = False
        Me.Label15.AutoSize = True
        Me.Label15.Location = New System.Drawing.Point(606, 633)
        Me.Label15.Name = "Label15"
        Me.Label15.Size = New System.Drawing.Size(42, 13)
        Me.Label15.TabIndex = 38
        Me.Label15.Text = "Time :"
        Me.Label15.Visible = False
        Me.Timer1.Interval = 8000
        Me.Label16.AutoSize = True
        Me.Label16.Location = New System.Drawing.Point(556, 705)
        Me.Label16.Name = "Label16"
        Me.Label16.Size = New System.Drawing.Size(0, 13)
        Me.Label16.TabIndex = 39
        Me.Label16.Visible = False
        Me.TextBox4.Location = New System.Drawing.Point(684, 651)
        Me.TextBox4.Name = "TextBox4"
        Me.TextBox4.Size = New System.Drawing.Size(66, 20)
        Me.TextBox4.TabIndex = 40
        Me.TextBox4.Visible = False
        Me.Label17.AutoSize = True
        Me.Label17.Location = New System.Drawing.Point(606, 658)
        Me.Label17.Name = "Label17"
        Me.Label17.Size = New System.Drawing.Size(42, 13)
        Me.Label17.TabIndex = 41
        Me.Label17.Text = "Date :"
        Me.Label17.Visible = False
        Me.Label19.AutoSize = True
        Me.Label19.Font = New System.Drawing.Font("Microsoft Sans
Serif", 14.25!, System.Drawing.FontStyle.Bold,
System.Drawing.GraphicsUnit.Point, CType(0, Byte))
        Me.Label19.ForeColor = System.Drawing.Color.Red
        Me.Label19.Location = New System.Drawing.Point(370, 444)
        Me.Label19.Name = "Label19"
        Me.Label19.Size = New System.Drawing.Size(113, 24)
        Me.Label19.TabIndex = 43
        Me.Label19.Text = "UPDATING"
        Me.Label19.Visible = False
        Me.Button1.Location = New System.Drawing.Point(661, 725)
        Me.Button1.Name = "Button1"
        Me.Button1.Size = New System.Drawing.Size(105, 27)
        Me.Button1.TabIndex = 44
        Me.Button1.Text = "Exit"
        Me.Button1.UseVisualStyleBackColor = True
        Me.Button1.Visible = False
        Me.Button2.Location = New System.Drawing.Point(551, 738)
        Me.Button2.Name = "Button2"
        Me.Button2.Size = New System.Drawing.Size(27, 21)
        Me.Button2.TabIndex = 45
```

```
        Me.Button2.Text = ">>"
        Me.Button2.UseVisualStyleBackColor = True
        Me.Button2.Visible = False
        Me.TextBox6.Location = New System.Drawing.Point(684, 600)
        Me.TextBox6.Name = "TextBox6"
        Me.TextBox6.Size = New System.Drawing.Size(66, 20)
        Me.TextBox6.TabIndex = 47
        Me.TextBox6.Visible = False
        Me.Label20.AutoSize = True
        Me.Label20.Location = New System.Drawing.Point(606, 607)
        Me.Label20.Name = "Label20"
        Me.Label20.Size = New System.Drawing.Size(57, 13)
        Me.Label20.TabIndex = 48
        Me.Label20.Text = "Tamper :"
        Me.Label20.Visible = False
        Me.ListView1.Columns.AddRange(New
System.Windows.Forms.ColumnHeader() {Me.ID, Me.Full_Name, Me.Zone,
Me.Status, Me.A_Date, Me.A_Time})
        Me.ListView1.Font = New System.Drawing.Font("Microsoft Sans
Serif", 8.25!, System.Drawing.FontStyle.Regular,
System.Drawing.GraphicsUnit.Point, CType(0, Byte))
        Me.ListView1.FullRowSelect = True
        Me.ListView1.Location = New System.Drawing.Point(36, 100)
        Me.ListView1.MultiSelect = False
        Me.ListView1.Name = "ListView1"
        Me.ListView1.Size = New System.Drawing.Size(555, 246)
        Me.ListView1.TabIndex = 49
        Me.ListView1.UseCompatibleStateImageBehavior = False
        Me.ListView1.View = System.Windows.Forms.View.Details

        Me.ID.Text = "Tag ID"
        Me.ID.Width = 57
        Me.Full_Name.Text = "Course Name"
        Me.Full_Name.TextAlign =
System.Windows.Forms.HorizontalAlignment.Center
        Me.Full_Name.Width = 150

        Me.Zone.Text = "Course Code"
        Me.Zone.TextAlign =
System.Windows.Forms.HorizontalAlignment.Center
        Me.Zone.Width = 90
        Me.Status.Text = "Status"
        Me.Status.TextAlign =
System.Windows.Forms.HorizontalAlignment.Center
        Me.Status.Width = 95
        Me.A_Date.Text = "Date"
        Me.A_Date.TextAlign =
System.Windows.Forms.HorizontalAlignment.Center
        Me.A_Date.Width = 80
        Me.A_Time.Text = "Time"
        Me.A_Time.TextAlign =
System.Windows.Forms.HorizontalAlignment.Center
        Me.A_Time.Width = 80
        Me.lblOpName.AutoSize = True
        Me.lblOpName.BackColor = System.Drawing.Color.Azure
```

```
        Me.lblOpName.Location = New System.Drawing.Point(96, 152)
        Me.lblOpName.Name = "lblOpName"
        Me.lblOpName.Size = New System.Drawing.Size(0, 13)
        Me.lblOpName.TabIndex = 53
        Me.commStatus.AutoSize = True
        Me.commStatus.BackColor = System.Drawing.Color.Red
        Me.commStatus.Font = New System.Drawing.Font("Franklin Gothic
Book", 15.75!, System.Drawing.FontStyle.Bold,
System.Drawing.GraphicsUnit.Point, CType(0, Byte))
        Me.commStatus.Location = New System.Drawing.Point(642, 521)
        Me.commStatus.Name = "commStatus"
        Me.commStatus.Size = New System.Drawing.Size(50, 26)
        Me.commStatus.TabIndex = 52
        Me.commStatus.Text = "OFF"
        Me.lbl_Oname.AutoSize = True
        Me.lbl_Oname.Location = New System.Drawing.Point(152, 154)
        Me.lbl_Oname.Name = "lbl_Oname"
        Me.lbl_Oname.Size = New System.Drawing.Size(0, 13)
        Me.lbl_Oname.TabIndex = 51
        Me.lbl_Operator.AutoSize = True
        Me.lbl_Operator.BackColor = System.Drawing.Color.Crimson
        Me.lbl_Operator.BorderStyle =
System.Windows.Forms.BorderStyle.Fixed3D
        Me.lbl_Operator.Font = New System.Drawing.Font("Franklin
Gothic Book", 9.75!, System.Drawing.FontStyle.Bold,
System.Drawing.GraphicsUnit.Point, CType(0, Byte))
        Me.lbl_Operator.ForeColor = System.Drawing.Color.FloralWhite
        Me.lbl_Operator.Location = New System.Drawing.Point(24, 28)
        Me.lbl_Operator.Name = "lbl_Operator"
        Me.lbl_Operator.Size = New System.Drawing.Size(74, 19)
        Me.lbl_Operator.TabIndex = 50
        Me.lbl_Operator.Text = "Operator :"
        Me.BtnOpList.BackColor =
System.Drawing.Color.FromArgb(CType(CType(255, Byte), Integer),
CType(CType(192, Byte), Integer), CType(CType(255, Byte), Integer))
        Me.BtnOpList.Font = New System.Drawing.Font("Franklin Gothic
Book", 11.25!, System.Drawing.FontStyle.Bold,
System.Drawing.GraphicsUnit.Point, CType(0, Byte))
        Me.BtnOpList.ForeColor = System.Drawing.Color.Black
        Me.BtnOpList.Location = New System.Drawing.Point(615, 116)
        Me.BtnOpList.Name = "BtnOpList"
        Me.BtnOpList.Size = New System.Drawing.Size(87, 31)
        Me.BtnOpList.TabIndex = 59
        Me.BtnOpList.Text = "Staff List"
        Me.BtnOpList.UseVisualStyleBackColor = False
        Me.BtnInmLst.BackColor =
System.Drawing.Color.FromArgb(CType(CType(255, Byte), Integer),
CType(CType(192, Byte), Integer), CType(CType(255, Byte), Integer))
        Me.BtnInmLst.Font = New System.Drawing.Font("Franklin Gothic
Book", 11.25!, System.Drawing.FontStyle.Bold,
System.Drawing.GraphicsUnit.Point, CType(0, Byte))
        Me.BtnInmLst.ForeColor = System.Drawing.Color.Black
        Me.BtnInmLst.Location = New System.Drawing.Point(615, 159)
        Me.BtnInmLst.Name = "BtnInmLst"
        Me.BtnInmLst.Size = New System.Drawing.Size(86, 31)
```

```
        Me.BtnInmLst.TabIndex = 58
        Me.BtnInmLst.Text = "Doc List"
        Me.BtnInmLst.UseVisualStyleBackColor = False
        Me.BtnStart.BackColor =
System.Drawing.Color.FromArgb(CType(CType(255, Byte), Integer),
CType(CType(192, Byte), Integer), CType(CType(255, Byte), Integer))
        Me.BtnStart.Font = New System.Drawing.Font("Franklin Gothic
Book", 11.25!, System.Drawing.FontStyle.Bold,
System.Drawing.GraphicsUnit.Point, CType(0, Byte))
        Me.BtnStart.ForeColor = System.Drawing.Color.Black
        Me.BtnStart.Location = New System.Drawing.Point(615, 73)
        Me.BtnStart.Name = "BtnStart"
        Me.BtnStart.Size = New System.Drawing.Size(86, 31)
        Me.BtnStart.TabIndex = 57
        Me.BtnStart.Text = "Start"
        Me.BtnStart.UseVisualStyleBackColor = False
        Me.BtnAddOfficer.BackColor =
System.Drawing.Color.FromArgb(CType(CType(255, Byte), Integer),
CType(CType(192, Byte), Integer), CType(CType(255, Byte), Integer))
        Me.BtnAddOfficer.Font = New System.Drawing.Font("Franklin
Gothic Book", 11.25!, System.Drawing.FontStyle.Bold,
System.Drawing.GraphicsUnit.Point, CType(0, Byte))
        Me.BtnAddOfficer.ForeColor = System.Drawing.Color.Black
        Me.BtnAddOfficer.Location = New System.Drawing.Point(615,
269)
        Me.BtnAddOfficer.Name = "BtnAddOfficer"
        Me.BtnAddOfficer.Size = New System.Drawing.Size(87, 31)
        Me.BtnAddOfficer.TabIndex = 56
        Me.BtnAddOfficer.Text = "Add Staff" &
Global.Microsoft.VisualBasic.ChrW(13) &
Global.Microsoft.VisualBasic.ChrW(10)
        Me.BtnAddOfficer.UseVisualStyleBackColor = False
        Me.BtnAddInmate.BackColor =
System.Drawing.Color.FromArgb(CType(CType(255, Byte), Integer),
CType(CType(192, Byte), Integer), CType(CType(255, Byte), Integer))
        Me.BtnAddInmate.Font = New System.Drawing.Font("Franklin
Gothic Book", 11.25!, System.Drawing.FontStyle.Bold,
System.Drawing.GraphicsUnit.Point, CType(0, Byte))
        Me.BtnAddInmate.ForeColor = System.Drawing.Color.Black
        Me.BtnAddInmate.Location = New System.Drawing.Point(615, 315)
        Me.BtnAddInmate.Name = "BtnAddInmate"
        Me.BtnAddInmate.Size = New System.Drawing.Size(87, 31)
        Me.BtnAddInmate.TabIndex = 55
        Me.BtnAddInmate.Text = "Add Doc"
        Me.BtnAddInmate.UseVisualStyleBackColor = False
        Me.BtnLogOut.BackColor =
System.Drawing.Color.FromArgb(CType(CType(255, Byte), Integer),
CType(CType(192, Byte), Integer), CType(CType(255, Byte), Integer))
        Me.BtnLogOut.Font = New System.Drawing.Font("Franklin Gothic
Book", 11.25!, System.Drawing.FontStyle.Bold,
System.Drawing.GraphicsUnit.Point, CType(0, Byte))
        Me.BtnLogOut.ForeColor = System.Drawing.Color.Black
        Me.BtnLogOut.Location = New System.Drawing.Point(620, 471)
        Me.BtnLogOut.Name = "BtnLogOut"
        Me.BtnLogOut.Size = New System.Drawing.Size(81, 31)
```

```
        Me.BtnLogOut.TabIndex = 54
        Me.BtnLogOut.Text = "Log Out"
        Me.BtnLogOut.UseVisualStyleBackColor = False
        Me.TmrCall.Interval = 9000
        Me.Button3.BackColor =
System.Drawing.Color.FromArgb(CType(CType(255, Byte), Integer),
CType(CType(192, Byte), Integer), CType(CType(255, Byte), Integer))
        Me.Button3.Font = New System.Drawing.Font("Franklin Gothic
Book", 11.25!, System.Drawing.FontStyle.Bold,
System.Drawing.GraphicsUnit.Point, CType(0, Byte))
        Me.Button3.ForeColor = System.Drawing.Color.Black
        Me.Button3.Location = New System.Drawing.Point(615, 204)
        Me.Button3.Name = "Button3"
        Me.Button3.Size = New System.Drawing.Size(86, 52)
        Me.Button3.TabIndex = 60
        Me.Button3.Text = "Search Doc"
        Me.Button3.UseVisualStyleBackColor = False
        Me.Label18.AutoSize = True
        Me.Label18.BackColor = System.Drawing.Color.BlanchedAlmond
        Me.Label18.Font = New System.Drawing.Font("Franklin Gothic
Book", 12.0!, CType((System.Drawing.FontStyle.Bold Or
System.Drawing.FontStyle.Italic), System.Drawing.FontStyle),
System.Drawing.GraphicsUnit.Point, CType(0, Byte))
        Me.Label18.Location = New System.Drawing.Point(103, 27)
        Me.Label18.Name = "Label18"
        Me.Label18.Size = New System.Drawing.Size(73, 21)
        Me.Label18.TabIndex = 61
        Me.Label18.Text = "Label18"
        Me.AutoScaleBaseSize = New System.Drawing.Size(6, 13)
        Me.BackgroundImage =
CType(resources.GetObject("$this.BackgroundImage"),
System.Drawing.Image)
        Me.BackgroundImageLayout =
System.Windows.Forms.ImageLayout.Stretch
        Me.ClientSize = New System.Drawing.Size(717, 587)
        Me.Controls.Add(Me.Label18)
        Me.Controls.Add(Me.Button3)
        Me.Controls.Add(Me.BtnOpList)
        Me.Controls.Add(Me.BtnInmLst)
        Me.Controls.Add(Me.BtnStart)
        Me.Controls.Add(Me.BtnAddOfficer)
        Me.Controls.Add(Me.BtnAddInmate)
        Me.Controls.Add(Me.BtnLogOut)
        Me.Controls.Add(Me.lblOpName)
        Me.Controls.Add(Me.commStatus)
        Me.Controls.Add(Me.lbl_Oname)
        Me.Controls.Add(Me.lbl_Operator)
        Me.Controls.Add(Me.ListView1)
        Me.Controls.Add(Me.PictureBox2)
        Me.Controls.Add(Me.Label20)
        Me.Controls.Add(Me.rfOpenSocketButton)
        Me.Controls.Add(Me.PictureBox1)
        Me.Controls.Add(Me.TextBox6)
        Me.Controls.Add(Me.Button2)
        Me.Controls.Add(Me.Button1)
```

```
        Me.Controls.Add(Me.Label19)
        Me.Controls.Add(Me.QueryButton)
        Me.Controls.Add(Me.Label17)
        Me.Controls.Add(Me.rfScanButton)
        Me.Controls.Add(Me.TextBox4)
        Me.Controls.Add(Me.Label16)
        Me.Controls.Add(Me.Label15)
        Me.Controls.Add(Me.TextBox3)
        Me.Controls.Add(Me.Label14)
        Me.Controls.Add(Me.Label13)
        Me.Controls.Add(Me.TextBox2)
        Me.Controls.Add(Me.TextBox1)
        Me.Controls.Add(Me.GroupBox4)
        Me.Controls.Add(Me.LongIntervalCheckBox)
        Me.Controls.Add(Me.ClearButton)
        Me.Controls.Add(Me.IPListBox)
        Me.Controls.Add(Me.Label11)
        Me.Controls.Add(Me.FGenIDTextBox)
        Me.Controls.Add(Me.ReaderIDTextBox)
        Me.Controls.Add(Me.TagBox)
        Me.Controls.Add(Me.MsgArea)
        Me.Controls.Add(Me.Label6)
        Me.Controls.Add(Me.RdrCmdTypeComboBox)
        Me.Controls.Add(Me.Label5)
        Me.Controls.Add(Me.Label1)
        Me.Controls.Add(Me.GroupBox3)
        Me.Controls.Add(Me.GroupBox2)
        Me.Controls.Add(Me.GroupBox1)
        Me.Controls.Add(Me.Label4)
        Me.Controls.Add(Me.TagCmdTypeComboBox)
        Me.Controls.Add(Me.INVRadioButton)
        Me.Controls.Add(Me.ASTRadioButton)
        Me.Controls.Add(Me.ACCRadioButton)
        Me.Controls.Add(Me.Label3)
        Me.Controls.Add(Me.Label2)
        Me.Controls.Add(Me.SocketFlg)
        Me.Font = New System.Drawing.Font("Microsoft Sans Serif",
8.25!, System.Drawing.FontStyle.Bold,
System.Drawing.GraphicsUnit.Point, CType(0, Byte))
        Me.MaximizeBox = False
        Me.Name = "Form1"
        Me.StartPosition =
System.Windows.Forms.FormStartPosition.CenterScreen
        Me.Text = " D'TraXX Menu"
        Me.GroupBox1.ResumeLayout(False)
        Me.GroupBox6.ResumeLayout(False)
        Me.GroupBox5.ResumeLayout(False)
        Me.GroupBox5.PerformLayout()
        CType(Me.PictureBox2,
System.ComponentModel.ISupportInitialize).EndInit()
        CType(Me.PictureBox1,
System.ComponentModel.ISupportInitialize).EndInit()
        Me.GroupBox2.ResumeLayout(False)
        Me.GroupBox2.PerformLayout()
        Me.GroupBox3.ResumeLayout(False)
```

```
        Me.GroupBox4.ResumeLayout(False)
        Me.GroupBox4.PerformLayout()
        Me.ResumeLayout(False)
        Me.PerformLayout()

    End Sub

#End Region

    Private Sub Form1_Load(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles MyBase.Load

        System.Windows.Forms.Control.CheckForIllegalCrossThreadCalls
= False

        readerPort = Convert.ToUInt16(10001)
        commPort = Convert.ToUInt32(1)
        commBaud = Convert.ToUInt32(115200)

        ReaderEventHandler = New AW_API_NET.fReaderEvent(AddressOf
Me.OnReaderEvent)  'AddressOf Me.OnReaderEvent
        TagEventHandler = New AW_API_NET.fTagEvent(AddressOf
Me.OnTagEvent)

        registered = False
        checkPswd("E8696", 123)
    End Sub

    Private Sub Button1_Click(ByVal sender As System.Object, ByVal e
As System.EventArgs) Handles rfOpenButton.Click
        Dim iRet As Integer

        myPKTID = myPKTID + 1

        iRet = ActiveWaveAPI.rfOpen(commBaud, commPort)
        If (iRet = 0) Then
            AddMsg("Open Com Port:" + commPort.ToString() + " baud:"
+ commBaud.ToString() + " Successful")
        Else
            AddMsg("Open Com Port Failed")
        End If

        If registered = False Then
            ' Register reader callback handler
            ActiveWaveAPI.rfRegisterReaderEvent(ReaderEventHandler)
            AddMsg("ReaderEvent registered")

            ' Register tag callback handler
            ActiveWaveAPI.rfRegisterTagEvent(TagEventHandler)
            AddMsg("TagEvent registered")

            registered = True
        End If

    End Sub
```

```
    Private Sub Button2_Click(ByVal sender As System.Object, ByVal e
As System.EventArgs) Handles rfCloseButton.Click
        Dim iRet As Integer
        iRet = ActiveWaveAPI.rfClose()
        AddMsg("Close: " + iRet.ToString())

    End Sub

    Private Sub CheckBox1_CheckedChanged(ByVal sender As
System.Object, ByVal e As System.EventArgs)

    End Sub
    Private Sub AddMsg(ByVal msg As String)
        If msg.Length > 0 Then
            MsgArea.AppendText(vbCrLf & Now.ToString("hh:mm:ss") + ":
" & msg & vbCrLf)
        End If
        MsgArea.Invalidate()
    End Sub

    Private Sub Button3_Click(ByVal sender As System.Object, ByVal e
As System.EventArgs) Handles rfResetRdrButton.Click
        Dim iRet As Integer

        If (STDFGenCheckBox.Checked = True) Then
            MsgBox("Change baudrate to 115200", MsgBoxStyle.Critical,
"D'TraXX")
            Return
        End If

        myPKTID = myPKTID + 1
        If (RdrCmdTypeComboBox.Text.Equals("ALL_READERS")) Then
            iRet = ActiveWaveAPI.rfResetReader(UInt16.Parse(1),
UInt16.Parse(0), UInt16.Parse(0),
Convert.ToUInt16(AW_API_NET.APIConsts.ALL_READERS),
Convert.ToUInt16(myPKTID))
        Else
            If (ReaderIDTextBox.Text.Equals("")) Then
                MsgBox("No Reader ID", MsgBoxStyle.OkOnly, "Error
Msg")
                Return
            End If
            iRet = ActiveWaveAPI.rfResetReader(UInt16.Parse(1),
Convert.ToUInt16(ReaderIDTextBox.Text), UInt16.Parse(0),
Convert.ToUInt16(AW_API_NET.APIConsts.SPECIFIC_READER),
Convert.ToUInt16(myPKTID))
        End If
        AddMsg("ResetReader: " + iRet.ToString())
    End Sub

    Private Sub TagButton_Click(ByVal sender As System.Object, ByVal
e As System.EventArgs) Handles TagButton.Click
        Dim iRet As Integer
```

```
        Dim tagSelect As AW_API_NET.rfTagSelect_t
        Dim tagList(50) As UInt32
        Dim rdrID As UInt16
        Dim longInterval As Boolean
        Dim RdrCmdType As Integer

        If (STDFGenCheckBox.Checked = True) Then
            MsgBox("Change baudrate to 115200", MsgBoxStyle.Critical,
"D'TraXX")
            Return
        End If

        If (RdrCmdTypeComboBox.SelectedIndex = 0) Then
            If (ReaderIDTextBox.Text.Equals("")) Then
                MsgBox("No Reader ID", MsgBoxStyle.OkOnly, "Error
Msg")
                Return
            Else
                rdrID = Convert.ToUInt16(ReaderIDTextBox.Text)
            End If
        Else
            rdrID = UInt16.Parse(0)
        End If

        If (RdrCmdTypeComboBox.SelectedIndex = 0) Then
            RdrCmdType = ActiveWaveAPI.SPECIFIC_READER
        Else
            RdrCmdType = ActiveWaveAPI.ALL_READERS
        End If

        tagSelect.tagList = tagList

        If (TagBox.Text.Equals("")) Then
            If ((TagCmdTypeComboBox.SelectedIndex = 0) Or
(TagCmdTypeComboBox.SelectedIndex = 3)) Then
                MsgBox("No Tag ID", MsgBoxStyle.OkOnly, "Error Msg")
                Return
            Else
                tagSelect.tagList(0) = UInt32.Parse(0)
                tagSelect.numTags = Convert.ToUInt32(1)
            End If
        Else
            tagSelect.tagList(0) = Convert.ToUInt32(TagBox.Text)
            tagSelect.numTags = Convert.ToUInt32(1)
        End If

        If (TagCmdTypeComboBox.SelectedIndex = 0) Then
            tagSelect.selectType =
Convert.ToUInt32(AW_API_NET.APIConsts.RF_SELECT_TAG_ID)
        ElseIf (TagCmdTypeComboBox.SelectedIndex = 1) Then
            tagSelect.selectType =
Convert.ToUInt32(AW_API_NET.APIConsts.RF_SELECT_FIELD)
        ElseIf (TagCmdTypeComboBox.SelectedIndex = 2) Then
            tagSelect.selectType =
Convert.ToUInt32(AW_API_NET.APIConsts.RF_SELECT_TAG_TYPE)
```

```
        Else
            tagSelect.selectType =
Convert.ToUInt32(AW_API_NET.APIConsts.RF_SELECT_TAG_RANGE)
        End If

        If (ACCRadioButton.Checked) Then
            tagSelect.tagType = ActiveWaveAPI.ACCESS_TAG
        ElseIf (ASTRadioButton.Checked) Then
            tagSelect.tagType = ActiveWaveAPI.ASSET_TAG
        Else
            tagSelect.tagType = ActiveWaveAPI.INVENTORY_TAG
        End If

        If (myPKTID >= 223) Then
            myPKTID = 1
        Else
            myPKTID = myPKTID + 1
        End If

        If (LongIntervalCheckBox.Checked) Then
            longInterval = True
        Else
            longInterval = False
        End If

        iRet = ActiveWaveAPI.rfCallTags(UInt16.Parse(1), rdrID,
UInt16.Parse(0), UInt16.Parse(0), tagSelect, True, longInterval,
Convert.ToUInt16(RdrCmdType), Convert.ToUInt16(myPKTID))
        AddMsg("CallTags: " + iRet.ToString())
    End Sub
    Private Function checkPswd(ByVal name As String, ByVal pswd As
String) As Boolean

        Dim found As Boolean = False
        Dim rows As Integer = 0
        Dim myConnString As String = "Database= dtraxx_db;Data
Source=localhost;User Id=root;Password="
        Dim cn As New MySqlConnection(myConnString)
        Dim cmdGH As New MySqlCommand("SELECT * FROM staff WHERE id =
'" & name & "'AND Password = '" & pswd & "'", cn)
        Dim daGH As New MySqlDataAdapter
        Dim dsGH As New DataSet
        Dim dtGH As New DataTable
        Dim offName As String = ""

        cn.Open()

        Try
            daGH.SelectCommand = cmdGH
            daGH.Fill(dsGH, "employees")
            dtGH = dsGH.Tables("employees")
            rows = dtGH.Rows.Count()
        Catch ex As Exception
            MsgBox("Error: " & ex.Source & ": " & ex.Message,
MsgBoxStyle.OkOnly, "Connection Error !!")
```

```
        End Try

        If ConnectionState.Open Then
            cn.Close()
        End If

        If (rows > 0) Then
            found = True
            Dim namCmd As New MySqlCommand("SELECT name FROM staff
WHERE id = '" & name & "'", cn)
            Dim rdr As MySqlDataReader
            cn.Open()
            rdr = namCmd.ExecuteReader
            While rdr.Read
                offName = rdr.Item("name").ToString
            End While
            rdr.Close()
            cn.Close()
            ofName = offName
        Else
            found = False
        End If

        Return found

    End Function
    Private Function OnReaderEvent(ByVal readerEvent As
AW_API_NET.rfReaderEvent_t) As Integer

        Dim ipStr As String

        'Beep()
        If
readerEvent.eventType.Equals(Convert.ToUInt16(AW_API_NET.APIConsts.RF
_SCAN_NETWORK)) Then
            For i As Integer = 0 To readerEvent.ip.Length - 1
                ipStr += Convert.ToChar(readerEvent.ip(i))
            Next i
            IPListBox.Items.Add(ipStr)
        ElseIf
readerEvent.eventType.Equals(Convert.ToUInt16(AW_API_NET.APIConsts.RF
_OPEN_SOCKET)) Then
            For i As Integer = 0 To readerEvent.ip.Length - 1
                ipStr += Convert.ToChar(readerEvent.ip(i))
            Next i
            AddMsg("Socket Opened IP = " + ipStr)
        ElseIf
readerEvent.eventType.Equals(Convert.ToUInt16(AW_API_NET.APIConsts.RF
_CLOSE_SOCKET)) Then
            For i As Integer = 0 To readerEvent.ip.Length - 1
                ipStr += Convert.ToChar(readerEvent.ip(i))
            Next i
            AddMsg("Socket Closed IP = " + ipStr)

            Dim index As Integer
```

```
            If ipStr.Length > 0 Then
                index = IPListBox.FindStringExact(ipStr)
                If index >= 0 Then
                    IPListBox.Items.RemoveAt(index)
                End If
            End If
        ElseIf
readerEvent.eventType.Equals(Convert.ToUInt16(AW_API_NET.APIConsts.RF
_STD_FGEN_POWERUP)) Then
            FGenIDTextBox.Text = readerEvent.fGenerator.ToString()
            AddMsg("STD FGen Powered UP")
        ElseIf
readerEvent.eventType.Equals(Convert.ToUInt16(AW_API_NET.APIConsts.RF
_READER_POWERUP)) Then
            ReaderIDTextBox.Text = readerEvent.reader.ToString()
            AddMsg("Reader Powered UP")
        ElseIf
readerEvent.eventType.Equals(Convert.ToUInt16(AW_API_NET.APIConsts.RF
_QUERY_STD_FGEN)) Then
            Dim str As String

            AddMsg(readerEvent.eventType.ToString)
            AddMsg(AW_API_NET.APIConsts.RF_READER_POWERUP.ToString)

            AddMsg("STD FGEN Query __________")
            str = readerEvent.smartFgen.fsValue
            AddMsg("FS Value = " + str)
            str = readerEvent.smartFgen.txTime
            AddMsg("TX Time = " + str)
            str = readerEvent.smartFgen.waitTime
            AddMsg("Wait Time = " + str)
            str = readerEvent.smartFgen.assignRdr
            AddMsg("Assigned Rdr = " + str)
        ElseIf
readerEvent.eventType.Equals(Convert.ToUInt16(AW_API_NET.APIConsts.RF
_GET_RDR_FS)) Then
            FSTextBox.Text = readerEvent.smartFgen.fsValue.ToString()
            AddMsg("Reader ID:" + readerEvent.reader.ToString() + "
FS:" + readerEvent.smartFgen.fsValue.ToString())
        ElseIf
readerEvent.eventType.Equals(Convert.ToUInt16(AW_API_NET.APIConsts.RF
_SET_RDR_FS)) Then
            AddMsg("Reader FS was set successfuly")
        ElseIf
readerEvent.eventType.Equals(Convert.ToUInt16(AW_API_NET.APIConsts.RF
_SCAN_IP)) Then
            ipStr = GetStringIP(readerEvent.ip)
            If ipStr.Length > 0 Then
                If IPListBox.FindStringExact(ipStr) = -1 Then
                    IPListBox.Items.Add(ipStr)
                End If
                AddMsg("ScanIP IP=" + ipStr)
            End If
        End If
```

```
        ReportReaderEvent(readerEvent)

        Return 0
    End Function

    Private Function OnTagEvent(ByVal tagEvent As
AW_API_NET.rfTagEvent_t) As Integer

        Dim ipStr As String

        'Beep()

        If
tagEvent.eventType.Equals(Convert.ToUInt16(AW_API_NET.APIConsts.RF_TA
G_READ)) Then
            Dim str As String
            Dim n As Integer
            n = CInt(Convert.ToInt16(tagEvent.tag.dataLen))
            For i As Integer = 0 To n - 1
                str = tagEvent.tag.data(i)
                ipStr += str + " "
            Next
            AddMsg("Data :" + ipStr + vbCrLf)
        End If

        ReportTagEvent(tagEvent)

        Return 0
    End Function

    Private Sub ReportReaderEvent(ByRef readerEvent As
AW_API_NET.rfReaderEvent_t)
        Dim msg As String

        msg = "eventType " + readerEvent.eventType.ToString() +
vbCrLf _
            + vbTab + "errorStatus = " +
readerEvent.errorStatus.ToString() + vbCrLf _
            + vbTab + "host = " + readerEvent.host.ToString() +
vbCrLf _
            + vbTab + "reader = " + readerEvent.reader.ToString() +
vbCrLf _
            + vbTab + "fGenerator = " +
readerEvent.fGenerator.ToString() + vbCrLf _
            + vbTab + "cmdType = " + readerEvent.cmdType.ToString() +
vbCrLf _
            + vbTab + "eventStatus = " +
readerEvent.eventStatus.ToString() + vbCrLf _
            + vbTab + "pktID = " + readerEvent.pktID.ToString() +
vbCrLf
        If readerEvent.eventType.ToString() = "163" Then
            Beep()
            PictureBox1.Visible = True
        End If
        If readerEvent.eventType.ToString() = "164" Then
```

```
            Beep()
            PictureBox2.Visible = True
        End If
        AddMsg(msg)
    End Sub

    Private Sub ReportTagEvent(ByRef tagEvent As
AW_API_NET.rfTagEvent_t)
        Dim msg As String

        Dim rowsCheck As Integer = 0


        Dim cn As New MySqlConnection(strCn)

        '*******************************


        msg = "eventType " + tagEvent.eventType.ToString() + vbCrLf _
            + vbTab + "errorStatus = " +
tagEvent.errorStatus.ToString() + vbCrLf _
            + vbTab + "host = " + tagEvent.host.ToString() + vbCrLf _
            + vbTab + "reader = " + tagEvent.reader.ToString() +
vbCrLf _
            + vbTab + "fGenerator = " +
tagEvent.fGenerator.ToString() + vbCrLf _
            + vbTab + "eventStatus = " +
tagEvent.eventStatus.ToString() + vbCrLf _
            + vbTab + "cmdType = " + tagEvent.cmdType.ToString() +
vbCrLf _
            + vbTab + "RSSI = " + tagEvent.RSSI.ToString() + vbCrLf _
            + vbTab + "tagID = " + tagEvent.tag.id.ToString() +
vbCrLf _
            + vbTab + "tagType = " + tagEvent.tag.tagType.ToString()
+ vbCrLf _
            + vbTab + "pktID = " + tagEvent.pktID.ToString() + vbCrLf

        AddMsg(msg)
        If tagEvent.eventType.ToString() = "164" Then
            PictureBox2.Visible = True
        End If
        If tagEvent.eventType.ToString() = "143" Or
tagEvent.eventType.ToString() = "132" Then
            TextBox1.Text = tagEvent.reader.ToString()
            TextBox2.Text = tagEvent.tag.id.ToString()
            TextBox3.Text = TimeOfDay
            TextBox4.Text = DateString
        End If
        ReportTagt(tagEvent.tag)

        If ConnectionState.Open Then
            cn.Close()
        End If
```

```
        Dim id As String = TextBox2.Text.ToString
        Dim room As String = TextBox1.Text.ToString
        Dim tarikh As String = TextBox4.Text.ToString
        Dim time As String = TextBox3.Text.ToString
        ' If room = 0 Then Label19.Visible = False

'*************************************************************
        Dim cmdCheck As New MySqlCommand("SELECT * FROM readers WHERE
ReaderID = '" & id & "'", cn)
        Dim daCheck As New MySqlDataAdapter
        Dim dsCheck As New DataSet
        Dim dtCheck As New DataTable

        cn.Open()

           Try
            With daCheck
                .SelectCommand = cmdCheck
                .Fill(dsCheck, "searchresult")
            End With
            dtCheck = dsCheck.Tables("searchresult")
            rowsCheck = dtCheck.Rows.Count()


        Catch ex As Exception
            MsgBox("Error: " & ex.Source & ": " & ex.Message,
MsgBoxStyle.OkOnly, "Connection Error !!")
        End Try
        If ConnectionState.Open Then
            cn.Close()
        End If


        If tagEvent.eventType.ToString() = "143" And room <> "0" Then
            If rowsCheck = 0 Then

                cn.Open()

                Dim cmdIns As New MySqlCommand("INSERT INTO readers
(ReaderID, RoomNo,Date,Time) VALUES ('" & id & "','" & room & "','" &
tarikh & "','" & time & "')", cn)
                cmdIns.ExecuteNonQuery()
                If ConnectionState.Open Then
                    cn.Close()
                End If
                Label16.Text = id + "/" + Name + " has been
registered successfully"
            Else
   Label16.Text = id + " is already registered in the database"
            End If
        End If

    End Sub
```

```
    Private Sub ReportTagt(ByRef tagt As AW_API_NET.rfTag_t)
        Dim msg As String

        msg = "ID " + tagt.id.ToString() + vbCrLf _
            + vbTab + "Type = " + tagt.tagType.ToString() + vbCrLf _
            + vbTab + "Status = " + tagt.status.ToString() + vbCrLf
        AddMsg(msg)
        ReportTagStatus(tagt.status)
    End Sub
    Private Sub ReportTagStatus(ByRef tagstatus As
AW_API_NET.rfTagStatus_t)
        Dim msg As String

        msg = "Batter Low = " + tagstatus.batteryLow.ToString() +
vbCrLf _
            + vbTab + "Tamper Switch = " +
tagstatus.tamperSwitch.ToString() + vbCrLf _
            + vbTab + "Enable = " + tagstatus.enabled.ToString() +
vbCrLf
        AddMsg(msg)
        If tagstatus.tamperSwitch.ToString() = True Then
            TextBox6.Text = "Open"
        Else
            TextBox6.Text = "Close"
        End If

    End Sub
    Private Sub QueryButton_Click(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles QueryButton.Click
        Dim iRet As Integer
        Dim tagSelect As AW_API_NET.rfTagSelect_t
        Dim tagList(50) As UInt32
        Dim rdrID As UInt16
        Dim longInterval As Boolean
        Dim RdrCmdType As Integer
        Label19.Visible = True
        If (STDFGenCheckBox.Checked = True) Then
            MsgBox("Change baudrate to 115200", MsgBoxStyle.Critical,
"D'TraXX")
            Return
        End If

        If (RdrCmdTypeComboBox.SelectedIndex = 0) Then
            If (ReaderIDTextBox.Text.Equals("")) Then
                MsgBox("No Reader ID", MsgBoxStyle.OkOnly, "Error
Msg")
                Return
            Else
                rdrID = Convert.ToUInt16(ReaderIDTextBox.Text)
            End If
        Else
            rdrID = UInt16.Parse(0)
        End If

        tagSelect.tagList = tagList
```

```
        If (TagBox.Text.Equals("")) Then
            If ((TagCmdTypeComboBox.SelectedIndex = 0) Or
(TagCmdTypeComboBox.SelectedIndex = 3)) Then
                MsgBox("No Tag ID", MsgBoxStyle.OkOnly, "Error Msg")
                Return
            Else
                tagSelect.tagList(0) = UInt32.Parse(0)
                tagSelect.numTags = Convert.ToUInt32(1)
            End If
        Else
            tagSelect.tagList(0) = Convert.ToUInt32(TagBox.Text)
            tagSelect.numTags = Convert.ToUInt32(1)
        End If

        If (RdrCmdTypeComboBox.SelectedIndex = 0) Then
            RdrCmdType = ActiveWaveAPI.SPECIFIC_READER
        Else
            RdrCmdType = ActiveWaveAPI.ALL_READERS
        End If

        If (TagCmdTypeComboBox.SelectedIndex = 0) Then
            tagSelect.selectType =
Convert.ToUInt32(AW_API_NET.APIConsts.RF_SELECT_TAG_ID)
        ElseIf (TagCmdTypeComboBox.SelectedIndex = 1) Then
            tagSelect.selectType =
Convert.ToUInt32(AW_API_NET.APIConsts.RF_SELECT_FIELD)
        ElseIf (TagCmdTypeComboBox.SelectedIndex = 2) Then
            tagSelect.selectType =
Convert.ToUInt32(AW_API_NET.APIConsts.RF_SELECT_TAG_TYPE)
        Else
            tagSelect.selectType =
Convert.ToUInt32(AW_API_NET.APIConsts.RF_SELECT_TAG_RANGE)
        End If

        If (ACCRadioButton.Checked) Then
            tagSelect.tagType = ActiveWaveAPI.ACCESS_TAG
        ElseIf (ASTRadioButton.Checked) Then
            tagSelect.tagType = ActiveWaveAPI.ASSET_TAG
        Else
            tagSelect.tagType = ActiveWaveAPI.INVENTORY_TAG
        End If

        If (myPKTID >= 223) Then
            myPKTID = 1
        Else
            myPKTID = myPKTID + 1
        End If

        If (LongIntervalCheckBox.Checked) Then
            longInterval = True
        Else
            longInterval = False
        End If
```

```
        iRet = ActiveWaveAPI.rfQueryTags(UInt16.Parse(1), rdrID,
UInt16.Parse(0), tagSelect, True, longInterval,
Convert.ToUInt16(RdrCmdType), Convert.ToUInt16(myPKTID))
        AddMsg("QueryTags: " + iRet.ToString())
        Timer1.Enabled = True
    End Sub

    Private Sub SocketFlg_CheckedChanged(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
SocketFlg.CheckedChanged

        If (SocketFlg.Checked) Then
            rfScanButton.Enabled = True
            rfOpenButton.Text = "rfOpenSocket"
            rfCloseButton.Text = "rfCloseSocket"
        Else
            rfScanButton.Enabled = False
            rfOpenButton.Text = "rfOpen"
            rfCloseButton.Text = "rfClose"
        End If

    End Sub

    Private Sub rfScanButton_Click(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles rfScanButton.Click

        Dim iRet As Integer

        If (myPKTID >= 223) Then
            myPKTID = 1
        Else
            myPKTID = myPKTID + 1
        End If

        If registered = False Then

            ' Register reader callback handler
            ActiveWaveAPI.rfRegisterReaderEvent(ReaderEventHandler)
            AddMsg("ReaderEvent registered")

            ' Register tag callback handler
            ActiveWaveAPI.rfRegisterTagEvent(TagEventHandler)
            AddMsg("TagEvent registered")

            registered = True
        End If

        IPListBox.Items.Clear()
        iRet = ActiveWaveAPI.rfScanNetwork(Convert.ToUInt16(myPKTID))
        AddMsg("rfScanNetwork: " + iRet.ToString())

    End Sub

    Private Sub ClearButton_Click(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles ClearButton.Click
```

```
        MsgArea.Clear()
    End Sub

    Private Sub rfQueryRdrButton_Click(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles rfQueryRdrButton.Click

        Dim iRet As Integer
        Dim rdrID As UInt16

        If (STDFGenCheckBox.Checked = True) Then
            MsgBox("Change baudrate to 115200", MsgBoxStyle.Critical,
"D'TraXX")
            Return
        End If

        If (RdrCmdTypeComboBox.SelectedIndex = 0) Then
            If (ReaderIDTextBox.Text.Equals("")) Then
                MsgBox("No Reader ID", MsgBoxStyle.OkOnly, "Error
Msg")
                Return
            Else
                rdrID = Convert.ToUInt16(ReaderIDTextBox.Text)
            End If
        Else
            rdrID = UInt16.Parse(0)
        End If



        If (myPKTID >= 223) Then
            myPKTID = 1
        Else
            myPKTID = myPKTID + 1
        End If

        If (RdrCmdTypeComboBox.SelectedIndex = 0) Then
            iRet = ActiveWaveAPI.rfQueryReader(UInt16.Parse(1),
rdrID, UInt16.Parse(0),
Convert.ToUInt16(ActiveWaveAPI.SPECIFIC_READER),
Convert.ToUInt16(myPKTID))
        Else
            iRet = ActiveWaveAPI.rfQueryReader(UInt16.Parse(1),
rdrID, UInt16.Parse(0), Convert.ToUInt16(ActiveWaveAPI.ALL_READERS),
Convert.ToUInt16(myPKTID))
        End If
        AddMsg("rfQueryReader: " + iRet.ToString())

    End Sub

    Private Sub EnableTagButton_Click(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles EnableTagButton.Click

        Dim iRet As Integer
        Dim tagSelect As AW_API_NET.rfTagSelect_t
        Dim tagList(50) As UInt32
```

```
        Dim rdrID As UInt16
        Dim RdrCmdType As Integer
        Dim longInterval As Boolean

        If (STDFGenCheckBox.Checked = True) Then
            MsgBox("Change baudrate to 115200", MsgBoxStyle.Critical,
"D'TraXX")
            Return
        End If

        If (RdrCmdTypeComboBox.SelectedIndex = 0) Then
            If (ReaderIDTextBox.Text.Equals("")) Then
                MsgBox("No Reader ID", MsgBoxStyle.OkOnly, "Error
Msg")
                Return
            Else
                rdrID = Convert.ToUInt16(ReaderIDTextBox.Text)
            End If
        Else
            rdrID = UInt16.Parse(0)
        End If

        tagSelect.tagList = tagList

        If (TagBox.Text.Equals("")) Then
            If ((TagCmdTypeComboBox.SelectedIndex = 0) Or
(TagCmdTypeComboBox.SelectedIndex = 3)) Then
                MsgBox("No Tag ID", MsgBoxStyle.OkOnly, "Error Msg")
                Return
            Else
                tagSelect.tagList(0) = UInt32.Parse(0)
                tagSelect.numTags = Convert.ToUInt32(1)
            End If
        Else
            tagSelect.tagList(0) = Convert.ToUInt32(TagBox.Text)
            tagSelect.numTags = Convert.ToUInt32(1)
        End If

        If (RdrCmdTypeComboBox.SelectedIndex = 0) Then
            RdrCmdType = AW_API_NET.APIConsts.SPECIFIC_READER
        Else
            RdrCmdType = AW_API_NET.APIConsts.ALL_READERS
        End If

        If (TagCmdTypeComboBox.SelectedIndex = 0) Then
            tagSelect.selectType =
Convert.ToUInt32(AW_API_NET.APIConsts.RF_SELECT_TAG_ID)
        ElseIf (TagCmdTypeComboBox.SelectedIndex = 1) Then
            tagSelect.selectType =
Convert.ToUInt32(AW_API_NET.APIConsts.RF_SELECT_FIELD)
        ElseIf (TagCmdTypeComboBox.SelectedIndex = 2) Then
            tagSelect.selectType =
Convert.ToUInt32(AW_API_NET.APIConsts.RF_SELECT_TAG_TYPE)
        Else
```

```
            tagSelect.selectType =
Convert.ToUInt32(AW_API_NET.APIConsts.RF_SELECT_TAG_RANGE)
        End If

        If (ACCRadioButton.Checked) Then
            tagSelect.tagType = AW_API_NET.APIConsts.ACCESS_TAG
        ElseIf (ASTRadioButton.Checked) Then
            tagSelect.tagType = AW_API_NET.APIConsts.ASSET_TAG
        Else
            tagSelect.tagType = AW_API_NET.APIConsts.INVENTORY_TAG
        End If

        If (myPKTID >= 223) Then
            myPKTID = 1
        Else
            myPKTID = myPKTID + 1
        End If

        If (LongIntervalCheckBox.Checked) Then
            longInterval = True
        Else
            longInterval = False
        End If

        iRet = ActiveWaveAPI.rfEnableTags(UInt16.Parse(1), rdrID,
UInt16.Parse(0), tagSelect, True, True, longInterval,
Convert.ToUInt16(RdrCmdType), Convert.ToUInt16(myPKTID))
        AddMsg("EnableTags: " + iRet.ToString())

    End Sub

    Private Sub ReadTagButton_Click(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles ReadTagButton.Click

        Dim iRet As Integer
        Dim tagSelect As AW_API_NET.rfTagSelect_t
        Dim tagList(50) As UInt32
        Dim rdrID As UInt16
        Dim RdrCmdType As Integer
        Dim longInterval As Boolean
        Dim address As Integer
        Dim len As UInt16

        If (STDFGenCheckBox.Checked = True) Then
            MsgBox("Change baudrate to 115200", MsgBoxStyle.Critical,
"D'TraXX")
            Return
        End If

        If (RdrCmdTypeComboBox.SelectedIndex = 0) Then
            If (ReaderIDTextBox.Text.Equals("")) Then
                MsgBox("No Reader ID", MsgBoxStyle.OkOnly, "Error
Msg")
                Return
            Else
```

```
                rdrID = Convert.ToUInt16(ReaderIDTextBox.Text)
            End If
        Else
            rdrID = UInt16.Parse(0)
        End If

        tagSelect.tagList = tagList

        If (TagBox.Text.Equals("")) Then
            If ((TagCmdTypeComboBox.SelectedIndex = 0) Or
(TagCmdTypeComboBox.SelectedIndex = 3)) Then
                MsgBox("No Tag ID", MsgBoxStyle.OkOnly, "Error Msg")
                Return
            Else
                tagSelect.tagList(0) = UInt32.Parse(0)
                tagSelect.numTags = Convert.ToUInt32(1)
            End If
        Else
            tagSelect.tagList(0) = Convert.ToUInt32(TagBox.Text)
            tagSelect.numTags = Convert.ToUInt32(1)
        End If

        address = CInt("&H" + AddressTextBox.Text)
        len = Convert.ToUInt16(LengthTextBox.Text)

        If (RdrCmdTypeComboBox.SelectedIndex = 0) Then
            RdrCmdType = ActiveWaveAPI.SPECIFIC_READER
        Else
            RdrCmdType = ActiveWaveAPI.ALL_READERS
        End If

        If (TagCmdTypeComboBox.SelectedIndex = 0) Then
            tagSelect.selectType =
Convert.ToUInt32(AW_API_NET.APIConsts.RF_SELECT_TAG_ID)
        ElseIf (TagCmdTypeComboBox.SelectedIndex = 1) Then
            tagSelect.selectType =
Convert.ToUInt32(AW_API_NET.APIConsts.RF_SELECT_FIELD)
        ElseIf (TagCmdTypeComboBox.SelectedIndex = 2) Then
            tagSelect.selectType =
Convert.ToUInt32(AW_API_NET.APIConsts.RF_SELECT_TAG_TYPE)
        Else
            tagSelect.selectType =
Convert.ToUInt32(AW_API_NET.APIConsts.RF_SELECT_TAG_RANGE)
        End If

        If (ACCRadioButton.Checked) Then
            tagSelect.tagType = AW_API_NET.APIConsts.ACCESS_TAG
        ElseIf (ASTRadioButton.Checked) Then
            tagSelect.tagType = AW_API_NET.APIConsts.ASSET_TAG
        Else
            tagSelect.tagType = AW_API_NET.APIConsts.INVENTORY_TAG
        End If

        If (myPKTID >= 223) Then
            myPKTID = 1
```

```
        Else
            myPKTID = myPKTID + 1
        End If

        If (LongIntervalCheckBox.Checked) Then
            longInterval = True
        Else
            longInterval = False
        End If

        iRet = ActiveWaveAPI.rfReadTags(UInt16.Parse(1), rdrID,
UInt16.Parse(0), tagSelect, Convert.ToUInt32(address), len, True,
longInterval, Convert.ToUInt16(RdrCmdType), UInt16.Parse(myPKTID))
        AddMsg("rfReadTags: " + iRet.ToString())
    End Sub

    Private Sub STDFGenCheckBox_CheckedChanged(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
STDFGenCheckBox.CheckedChanged

        Dim iRet As Integer

        If (STDFGenCheckBox.Checked) Then

            If SocketFlg.Checked Then
                ActiveWaveAPI.rfCloseSocket(readerIP,
Convert.ToUInt16(AW_API_NET.APIConsts.ALL_IPS))
                SocketFlg.Checked = False
                IPListBox.Items.Clear()
            Else
                ActiveWaveAPI.rfClose()
            End If

            rfScanButton.Enabled = False
            rfOpenButton.Enabled = False
            rfCloseButton.Enabled = False
            rfResetRdrButton.Enabled = False
            rfQueryRdrButton.Enabled = False
            TagButton.Enabled = False
            QueryButton.Enabled = False
            EnableTagButton.Enabled = False
            ReadTagButton.Enabled = False
            rfResetSmartFGenButton.Enabled = False
            rfQuerySTDFGenButton.Enabled = True

            iRet = ActiveWaveAPI.rfOpen(UInt32.Parse(9600), commPort)

            If (iRet = 0) Then
                AddMsg("Open Com Port:" + commPort.ToString() + "
baud:9600 Successful")
            Else
                AddMsg("Open Com Port Failed")
            End If

            If registered = False Then
```

```
                ' Register reader callback handler

ActiveWaveAPI.rfRegisterReaderEvent(ReaderEventHandler)
                AddMsg("ReaderEvent registered")

                ' Register tag callback handler
                ActiveWaveAPI.rfRegisterTagEvent(TagEventHandler)
                AddMsg("TagEvent registered")

                registered = True
            End If

        Else
            rfScanButton.Enabled = True
            rfOpenButton.Enabled = True
            rfCloseButton.Enabled = True
            rfResetRdrButton.Enabled = True
            rfQueryRdrButton.Enabled = True
            TagButton.Enabled = True
            QueryButton.Enabled = True
            EnableTagButton.Enabled = True
            ReadTagButton.Enabled = True
            rfResetSmartFGenButton.Enabled = True
            rfQuerySTDFGenButton.Enabled = False

            ActiveWaveAPI.rfClose()
            iRet = ActiveWaveAPI.rfOpen(UInt32.Parse(115200),
commPort)

            If (iRet = 0) Then
                AddMsg("Open Com Port:" + commPort.ToString() + "
baud:115200 Successful")
            Else
                AddMsg("Open Com Port Failed")
            End If

            If registered = False Then
                ' Register reader callback handler

ActiveWaveAPI.rfRegisterReaderEvent(ReaderEventHandler)
                AddMsg("ReaderEvent registered")

                ' Register tag callback handler
                ActiveWaveAPI.rfRegisterTagEvent(TagEventHandler)
                AddMsg("TagEvent registered")

                registered = True
            End If

        End If

    End Sub
```

```
    Private Sub rfQuerySTDFGenButton_Click(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
rfQuerySTDFGenButton.Click

        Dim iRet As Integer
        Dim fGenID As UInt16

        If (STDFGenCheckBox.Checked = False) Then
            MsgBox("Change baudrate to 9600", MsgBoxStyle.Critical,
"D'TraXX")
            Return
        End If

        If (FGenIDTextBox.Text.Equals("")) Then
            MsgBox("No FGen ID", MsgBoxStyle.OkOnly, "Error Msg")
            Return
        End If

        fGenID = Convert.ToUInt16(FGenIDTextBox.Text)

        If (myPKTID >= 223) Then
            myPKTID = 1
        Else
            myPKTID = myPKTID + 1
        End If

        iRet = ActiveWaveAPI.rfQuerySTDFGen(UInt16.Parse(1), fGenID,
Convert.ToUInt16(myPKTID))
        AddMsg("rfQuerySTDFGen: " + iRet.ToString())

    End Sub

    Private Sub BroadcastAllFGenCheckBox_CheckedChanged(ByVal sender
As System.Object, ByVal e As System.EventArgs)

    End Sub

    Private Sub rfResetSmartFGenButton_Click(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
rfResetSmartFGenButton.Click
        Dim iRet As Integer
        Dim fGenID As UInt16
        Dim rdrID As UInt16

        If (RdrCmdTypeComboBox.SelectedIndex = 0) Then
            If (ReaderIDTextBox.Text.Equals("")) Then
                MsgBox("No Reader ID", MsgBoxStyle.OkOnly, "Error
Msg")
                Return
            Else
                rdrID = Convert.ToUInt16(ReaderIDTextBox.Text)
            End If
        Else
            rdrID = UInt16.Parse(0)
        End If
```

```
        If (FGenIDTextBox.Text.Equals("")) Then
            MsgBox("No FGen ID", MsgBoxStyle.OkOnly, "Error Msg")
            Return
        End If

        If (STDFGenCheckBox.Checked = True) Then
            MsgBox("Change baudrate to 115200", MsgBoxStyle.Critical,
"D'TraXX")
            Return
        End If

        fGenID = Convert.ToUInt16(FGenIDTextBox.Text)

        If (myPKTID >= 223) Then
            myPKTID = 1
        Else
            myPKTID = myPKTID + 1
        End If

        Dim broadcast As Boolean = False
        If (BroadcastFGenCheckBox.Checked) Then
            broadcast = True
        End If

        If (RdrCmdTypeComboBox.SelectedIndex = 0) Then
            iRet = ActiveWaveAPI.rfResetSmartFGen(UInt16.Parse(1),
rdrID, UInt16.Parse(0), fGenID,
Convert.ToUInt16(AW_API_NET.APIConsts.SPECIFIC_READER), broadcast,
Convert.ToUInt16(myPKTID))
        Else
            iRet = ActiveWaveAPI.rfResetSmartFGen(UInt16.Parse(1),
rdrID, UInt16.Parse(0), fGenID,
Convert.ToUInt16(AW_API_NET.APIConsts.ALL_READERS), broadcast,
Convert.ToUInt16(myPKTID))
        End If

        AddMsg("rfResetSmartFGen: " + iRet.ToString())
    End Sub

    Private Sub rfSetReaderFSButton_Click(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
rfSetReaderFSButton.Click
        Dim iRet As Integer
        Dim rdrID As UInt16

        If (STDFGenCheckBox.Checked = True) Then
            MsgBox("Change baudrate to 115200", MsgBoxStyle.Critical,
"D'TraXX")
            Return
        End If

        If (FSTextBox.Text.Equals("")) Then
            MsgBox("No FS value", MsgBoxStyle.OkOnly, "Error Msg")
            Return
```

```
        End If

        If ((Convert.ToInt16(FSTextBox.Text) < 0) Or
(Convert.ToInt16(FSTextBox.Text) > 20)) Then
            MsgBox("FS value outside the limit(0-20)",
MsgBoxStyle.OkOnly, "Error Msg")
            Return
        End If

        If (RdrCmdTypeComboBox.SelectedIndex = 0) Then
            If (ReaderIDTextBox.Text.Equals("")) Then
                MsgBox("No Reader ID", MsgBoxStyle.OkOnly, "Error
Msg")
                Return
            Else
                rdrID = Convert.ToUInt16(ReaderIDTextBox.Text)
            End If
        Else
            rdrID = UInt16.Parse(0)
        End If

        If (myPKTID >= 223) Then
            myPKTID = 1
        Else
            myPKTID = myPKTID + 1
        End If

        If (RdrCmdTypeComboBox.SelectedIndex = 0) Then
            iRet = ActiveWaveAPI.rfSetReaderFS(UInt16.Parse(1),
rdrID, UInt16.Parse(0),
Convert.ToUInt16(AW_API_NET.APIConsts.RF_ABSOLUTE),
Convert.ToByte(FSTextBox.Text), False,
Convert.ToUInt16(ActiveWaveAPI.SPECIFIC_READER),
Convert.ToUInt16(myPKTID))
        Else
            iRet = ActiveWaveAPI.rfSetReaderFS(UInt16.Parse(1),
rdrID, UInt16.Parse(0),
Convert.ToUInt16(AW_API_NET.APIConsts.RF_ABSOLUTE),
Convert.ToByte(FSTextBox.Text), False,
Convert.ToUInt16(ActiveWaveAPI.ALL_READERS),
Convert.ToUInt16(myPKTID))
        End If
        AddMsg("rfSetReaderFS: " + iRet.ToString())
    End Sub

    Private Sub rfGetReaderFSButton_Click(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
rfGetReaderFSButton.Click
        Dim iRet As Integer
        Dim rdrID As UInt16

        If (STDFGenCheckBox.Checked = True) Then
            MsgBox("Change baudrate to 115200", MsgBoxStyle.Critical,
"D'TraXX")
            Return
```

```
        End If

        If (RdrCmdTypeComboBox.SelectedIndex = 0) Then
            If (ReaderIDTextBox.Text.Equals("")) Then
                MsgBox("No Reader ID", MsgBoxStyle.OkOnly, "Error
Msg")
                Return
            Else
                rdrID = Convert.ToUInt16(ReaderIDTextBox.Text)
            End If
        Else
            rdrID = UInt16.Parse(0)
        End If

        If (myPKTID >= 223) Then
            myPKTID = 1
        Else
            myPKTID = myPKTID + 1
        End If

        FSTextBox.Text = ""

        If (RdrCmdTypeComboBox.SelectedIndex = 0) Then
            iRet = ActiveWaveAPI.rfGetReaderFS(UInt16.Parse(1),
rdrID, UInt16.Parse(0),
Convert.ToUInt16(ActiveWaveAPI.SPECIFIC_READER),
Convert.ToUInt16(myPKTID))
        Else
            iRet = ActiveWaveAPI.rfGetReaderFS(UInt16.Parse(1),
rdrID, UInt16.Parse(0), Convert.ToUInt16(ActiveWaveAPI.ALL_READERS),
Convert.ToUInt16(myPKTID))
        End If
        AddMsg("rfGetReaderFS: " + iRet.ToString())

    End Sub

    Private Sub AllIPRadioButton_CheckedChanged(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
AllIPRadioButton.CheckedChanged
        If SpecificIPRadioButton.Checked Then
            rfScanButton.Enabled = False
            rfScanIPButton.Enabled = True
            IPTextBox.ReadOnly = False
        Else
            rfScanButton.Enabled = True
            rfScanIPButton.Enabled = False
            IPTextBox.ReadOnly = True
        End If
    End Sub

    Private Sub rfScanIPButton_Click(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles rfScanIPButton.Click
        Dim iRet As Integer
        Dim ip(20) As Byte
```

```
        If (myPKTID >= 223) Then
            myPKTID = 1
        Else
            myPKTID = myPKTID + 1
        End If

        If registered = False Then

            ' Register reader callback handler
            ActiveWaveAPI.rfRegisterReaderEvent(ReaderEventHandler)
            AddMsg("ReaderEvent registered")

            ' Register tag callback handler
            ActiveWaveAPI.rfRegisterTagEvent(TagEventHandler)
            AddMsg("TagEvent registered")

            registered = True
        End If

        If IPTextBox.TextLength = 0 Then
            AddMsg("No IP Address")
            Return
        End If

        For i As Integer = 0 To IPTextBox.Text.Length - 1
            ip(i) = Convert.ToByte(IPTextBox.Text.Chars(i))
        Next i


        iRet = ActiveWaveAPI.rfScanIP(ip, Convert.ToUInt16(myPKTID))
        AddMsg("rfScanIP: " + iRet.ToString())
    End Sub

    Private Sub rfOpenSocketButton_Click(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
rfOpenSocketButton.Click
        Dim iRet As Integer
        Dim ip(20) As Byte
        Dim cIP(20) As Char

        If PictureBox1.Visible = True Then
            PictureBox2.Visible = True
        End If


        myPKTID = myPKTID + 1
        If AllIPRadioButton.Checked Then
            iRet = ActiveWaveAPI.rfOpenSocket(readerIP, readerPort,
False, Convert.ToUInt16(AW_API_NET.APIConsts.ALL_IPS),
Convert.ToUInt16(myPKTID))
        Else
            Dim ipStr As String = IPTextBox.Text
            cIP = ipStr.ToCharArray(0, ipStr.Length)

            For i As Integer = 0 To IPTextBox.Text.Length - 1
```

```
                ip(i) = Convert.ToByte(IPTextBox.Text.Chars(i))
            Next i

            iRet = ActiveWaveAPI.rfOpenSocket(ip, readerPort, False,
Convert.ToUInt16(AW_API_NET.APIConsts.SPECIFIC_IP),
Convert.ToUInt16(myPKTID))
        End If

        If (iRet = 0) Then
            AddMsg("rfOpenSocket Successful. Return Code = " +
iRet.ToString())
        Else
            AddMsg("rfOpenSocket Failed Return Code = " +
iRet.ToString())
        End If

    End Sub

    Private Sub rfCloseSocketButton_Click(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
rfCloseSocketButton.Click
        Dim iRet As Integer
        Dim ip(20) As Byte
        Dim cIP(20) As Char

        myPKTID = myPKTID + 1
        If AllIPRadioButton.Checked Then
            iRet = ActiveWaveAPI.rfCloseSocket(readerIP,
Convert.ToUInt16(AW_API_NET.APIConsts.ALL_IPS))
            IPListBox.Items.Clear()
        Else
            Dim ipStr As String = IPTextBox.Text
            cIP = ipStr.ToCharArray(0, ipStr.Length)

            For i As Integer = 0 To IPTextBox.Text.Length - 1
                ip(i) = Convert.ToByte(IPTextBox.Text.Chars(i))
            Next i

            iRet = ActiveWaveAPI.rfCloseSocket(ip,
Convert.ToUInt16(AW_API_NET.APIConsts.SPECIFIC_IP))
        End If

        If (iRet = 0) Then
            AddMsg("rfCloseSocket Successful. Return Code = " +
iRet.ToString())
        Else
            AddMsg("rfCoseSocket Failed Return Code = " +
iRet.ToString())
        End If

    End Sub

    Public Function GetStringIP(ByVal ip As Byte()) As String

        Dim p As Integer
```

```
        Dim s As String
        Dim ct As Integer

        ct = 0
        p = 0
        s = ""
        While (Convert.ToBoolean((ct <= 3)) AndAlso
Convert.ToBoolean((p < 20)) AndAlso Convert.ToBoolean((ip(p) <> 0)))
            If ip(p) <> 46 Then
                s += Convert.ToString(ip(p) - 48) '- 48
                p += 1
            Else
                ct += 1
                p += 1
                s += "."
            End If

        End While
        Return s
    End Function

    Private Sub TagCmdTypeComboBox_SelectedIndexChanged(ByVal sender
As System.Object, ByVal e As System.EventArgs) Handles
TagCmdTypeComboBox.SelectedIndexChanged

    End Sub

    Private Sub Timer1_Tick(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Timer1.Tick
        Label19.Visible = True
        QueryButton_Click(sender, e)
        'Timer1.Enabled = False
    End Sub

    Private Sub GroupBox2_Enter(ByVal sender As System.Object, ByVal
e As System.EventArgs) Handles GroupBox2.Enter

    End Sub

    Private Sub SpecificIPRadioButton_CheckedChanged(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
SpecificIPRadioButton.CheckedChanged

    End Sub

    Private Sub MsgArea_TextChanged(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles MsgArea.TextChanged

    End Sub

    Private Sub Button1_Click_1(ByVal sender As System.Object, ByVal
e As System.EventArgs) Handles Button1.Click
        End
    End Sub
```

```
    Private Sub Button2_Click_1(ByVal sender As System.Object, ByVal
e As System.EventArgs) Handles Button2.Click
        If MsgArea.Visible = False Then
            MsgArea.Visible = True

        Else
            MsgArea.Visible = False
        End If
    End Sub
    Private Sub BtnStart_Click(ByVal sender As System.Object, ByVal e
As System.EventArgs) Handles BtnStart.Click

        'analize()
        checkdata()

        Me.commStatus.BackColor = System.Drawing.Color.Green

        Me.commStatus.Text = "ON"

    End Sub


    Private Sub checkdata()

        Dim check_id(1000) As String
        Dim check_name(1000) As String
        Dim check_room(1000) As String
        Dim i As Integer = 0
        Dim j As Integer
        Dim cn As New MySqlConnection(strCn)
        Dim cmd As New MySqlCommand("SELECT * FROM doc", cn)
        Dim da As New MySqlDataAdapter(cmd)
        Dim ds As New DataSet()
        Dim myReader As MySqlDataReader

        'Open Dtraxx database
        If cn.State = ConnectionState.Closed Then
            cn.Open()
        End If
        'read doclist database and store to buffer
        myReader = cmd.ExecuteReader()

        While myReader.Read
            i = i + 1
            check_id(i) = myReader.Item("tag_id").ToString
            check_name(i) = myReader.Item("Course_Name").ToString
            check_room(i) = myReader.Item("location").ToString

        End While

        If cn.State = ConnectionState.Open Then
            cn.Close()
        End If
        ' complete read doclist
```

```
        For j = 1 To i


            Dim cnchk As New MySqlConnection(strCn)
            Dim cmdchk As New MySqlCommand("SELECT * FROM readers",
cnchk)

            Dim dachk As New MySqlDataAdapter(cmdchk)
            Dim dschk As New DataSet()
            Dim myReaderchk As MySqlDataReader
            Dim found As Boolean


            If cnchk.State = ConnectionState.Closed Then
                cnchk.Open()
            End If

            myReaderchk = cmdchk.ExecuteReader()

            While myReaderchk.Read

                If check_id(j) =
myReaderchk.Item("ReaderID").ToString Then

                    found = True
                    Exit While
                Else
                    found = False
                End If

            End While

            If found = False Then

                cn.Open()
                Dim cmdIns As New MySqlCommand("UPDATE doc SET
doc_status = 'Not Available', date= '" & DateString.ToString & "',
time= '" & TimeString.ToString & "' WHERE tag_id = '" & check_id(j) &
"'", cn)
                cmdIns.ExecuteNonQuery()

                If ConnectionState.Open Then
                    cn.Close()
                End If

            Else

                cn.Open()
                Dim cmdIns As New MySqlCommand("UPDATE doc SET
doc_status = 'Available', date= '" & DateString.ToString & "', time=
'" & TimeString.ToString & "' WHERE tag_id = '" & check_id(j) & "'",
cn)
                cmdIns.ExecuteNonQuery()

                If ConnectionState.Open Then
```

```
                    cn.Close()
                End If

            End If

        Next j

        TmrCall.Enabled = True
        displaylistview()
        removealarm()
        clearReader()
    End Sub
    Public Sub removealarm()
        Dim cnchk As New MySqlConnection(strCn)
        Dim cmdchk As New MySqlCommand("SELECT * FROM readers",
cnchk)

        Dim dachk As New MySqlDataAdapter(cmdchk)
        Dim dschk As New DataSet()
        Dim myReaderchk As MySqlDataReader
        Dim found As Boolean
        Dim rm_id As String

        If cnchk.State = ConnectionState.Closed Then
            cnchk.Open()
        End If

        myReaderchk = cmdchk.ExecuteReader()

        While myReaderchk.Read


            rm_id = myReaderchk.Item("ReaderID").ToString

            '************************
            'Check if not in alarm
            Dim cnchk2 As New MySqlConnection(strCn)
            Dim cmdchk2 As New MySqlCommand("SELECT * FROM alarm",
cnchk2)

            Dim dachk2 As New MySqlDataAdapter(cmdchk)
            Dim dschk2 As New DataSet()
            Dim myReaderchk2 As MySqlDataReader
            Dim found2 As Boolean


            If cnchk2.State = ConnectionState.Closed Then
                cnchk2.Open()
            End If

            myReaderchk2 = cmdchk2.ExecuteReader()

            While myReaderchk2.Read
```

```
                If rm_id = myReaderchk2.Item("U_TagID").ToString Then
                    found2 = True
                    Exit While
                Else
                    found2 = False
                End If

            End While
            If cnchk2.State = ConnectionState.Open Then
                cnchk2.Close()
            End If

            '**********************
            If found2 = True Then
                Dim rowsCheck As Integer = 0
                Dim cn As New MySqlConnection(strCn)

                Dim cmdCheck As New MySqlCommand("SELECT FROM alarm
", cn)
                Dim daCheck As New MySqlDataAdapter
                Dim dsCheck As New DataSet
                Dim dtCheck As New DataTable


                If rowsCheck = 0 Then
                    'Open connection
                    cn.Open()

                    'Check for name and if no existing doc available,
add them
                    Dim cmdIns As New MySqlCommand("DELETE FROM alarm
WHERE U_TagID = '" & rm_id & "'", cn)
                    rowsCheck =
cmdIns.ExecuteReader.RecordsAffected()

                    If ConnectionState.Open Then
                        cn.Close()
                    End If

                Else

                End If


            End If


        End While
        If cnchk.State = ConnectionState.Closed Then
            cnchk.Open()
        End If
        displaylistview()

    End Sub
    Private Sub displaylistview()
```

```
        Dim i As Integer
        Dim cn As New MySqlConnection(strCn)
        'Dim cmd As New MySqlCommand("SELECT * FROM alarm ", cn)
        Dim cmd As New MySqlCommand("SELECT * FROM doc WHERE
doc_status = 'Available' ", cn)
        Dim da As New MySqlDataAdapter(cmd)
        Dim ds As New DataSet()
        Dim myReader As MySqlDataReader

        'Open dtraxx database
        If cn.State = ConnectionState.Closed Then
            cn.Open()
        End If
        'read doclist database and store to buffer
        ListView1.Items.Clear()

        myReader = cmd.ExecuteReader()

        While myReader.Read

            tagID =
ListView1.Items.Add(myReader.Item("tag_id").ToString)
            tagID.SubItems.Add(myReader.Item("Course_Name").ToString)
            tagID.SubItems.Add(myReader.Item("Course_Code").ToString)
            tagID.SubItems.Add(myReader.Item("doc_status").ToString)
            tagID.SubItems.Add(myReader.Item("date").ToString)
            tagID.SubItems.Add(myReader.Item("time").ToString)

        End While

        While i <= ListView1.Items.Count - 1
            If i Mod 2 = 0 Then
                ListView1.Items(i).BackColor = Color.BlanchedAlmond
            Else
                ListView1.Items(i).BackColor = Color.White
            End If
            i = i + 1
        End While

    End Sub



    Private Sub StoreAlarm()
        Dim i As Integer = 0
        Dim idchk As String
        Dim tag_id As String
        Dim ccode As String
        Dim cname As String
        Dim astatus As String
        Dim adate As String
        Dim atime As String
        Dim found As Boolean = False
```

```
        For i = 0 To ListView1.Items.Count - 1
            tagID = ListView1.Items(i)
            tag_id = tagID.SubItems(0).Text
            cname = tagID.SubItems(1).Text
            ccode = tagID.SubItems(2).Text
            astatus = tagID.SubItems(3).Text
            adate = tagID.SubItems(4).Text
            atime = tagID.SubItems(5).Text

            ' check data in alarm

            Dim cnchk As New MySqlConnection(strCn)
            Dim cmdchk As New MySqlCommand("SELECT * FROM alarm",
cnchk)

            Dim dachk As New MySqlDataAdapter(cmdchk)
            Dim dschk As New DataSet()
            Dim myReaderchk As MySqlDataReader



            If cnchk.State = ConnectionState.Closed Then
                cnchk.Open()
            End If

            'Display raw in listview (attendlist)
            myReaderchk = cmdchk.ExecuteReader()

            While myReaderchk.Read

                'str = myReader.getString(0)

                idchk = myReaderchk.Item("U_TagID").ToString
                If tag_id = idchk Then

                    found = True
                    Exit While
                Else
                    found = False
                End If

            End While



            If found = False Then



                Dim rowsCheck As Integer = 0
                Dim ID As String = 0

                Dim cn As New MySqlConnection(strCn)
```

```
                Dim cmdCheck As New MySqlCommand("SELECT * FROM alarm
WHERE U_TagID = '" & tag_id & "'", cn)
                Dim daCheck As New MySqlDataAdapter
                Dim dsCheck As New DataSet
                Dim dtCheck As New DataTable

                cn.Open()

                'Check from table (search)
                Try
                    With daCheck
                        .SelectCommand = cmdCheck
                        .Fill(dsCheck, "searchresult")
                    End With
                    dtCheck = dsCheck.Tables("searchresult")
                    rowsCheck = dtCheck.Rows.Count()

                    'If unable to connect, show error!
                Catch ex As Exception
                End Try

                'If already connected, close connection
                If ConnectionState.Open Then
                    cn.Close()
                End If


                If rowsCheck = 0 Then
                    'Open connection
                    cn.Open()

                    Dim cmdIns As New MySqlCommand("INSERT INTO alarm
(U_TagID, U_ccode,U_cname,U_Status) VALUES ('" & tag_id & "','" &
ccode & "','" & cname & "','" & astatus & "')", cn)
                    cmdIns.ExecuteNonQuery()
                    If cn.State = ConnectionState.Open Then
                        cn.Close()
                    End If

                    MsgBox(" Alarm .... " + tag_id + " is
unavailable", MsgBoxStyle.Critical, "D'TraXX")

                Else

                End If

            End If

                    If cnchk.State = ConnectionState.Open Then
                cnchk.Close()
            End If


        Next i
```

```
    End Sub
    Private Sub clearReader()

        Dim rowsCheck As Integer = 0


        Dim cn As New MySqlConnection(strCn)

        Dim cmdCheck As New MySqlCommand("DELETE FROM readers ", cn)
        Dim daCheck As New MySqlDataAdapter
        Dim dsCheck As New DataSet
        Dim dtCheck As New DataTable

        cn.Open()
        Try
            With daCheck
                .SelectCommand = cmdCheck
                .Fill(dsCheck, "searchresult")
            End With
            dtCheck = dsCheck.Tables("searchresult")
            rowsCheck = dtCheck.Rows.Count()
        Catch ex As Exception

        End Try

        If ConnectionState.Open Then
            cn.Close()
        End If


        If rowsCheck = 0 Then

            cn.Open()
Dim cmdIns As New MySqlCommand("DELETE FROM readers ", cn)
            rowsCheck = cmdIns.ExecuteReader.RecordsAffected()
            If ConnectionState.Open Then
                cn.Close()
            End If

        Else

        End If



    End Sub
    Public Sub alarmcheck(ByVal tagid As String)
        Dim rowsCheck As Integer = 0
        Dim cn As New MySqlConnection(strCn)
        Dim cmdCheck As New MySqlCommand("SELECT * FROM alarm WHERE
id = '" & tagid & "'", cn)
        Dim daCheck As New MySqlDataAdapter
        Dim dsCheck As New DataSet
        Dim dtCheck As New DataTable
```

```
        cn.Open()

        Try
            With daCheck
                .SelectCommand = cmdCheck
                .Fill(dsCheck, "searchresult")
            End With
            dtCheck = dsCheck.Tables("searchresult")
            rowsCheck = dtCheck.Rows.Count()
        Catch ex As Exception
            MsgBox("Error: " & ex.Source & ": " & ex.Message,
MsgBoxStyle.OkOnly, "Connection Error !!")
        End Try
        If ConnectionState.Open Then
            cn.Close()
        End If


        If rowsCheck = 0 Then
            cn.Open()
            Dim cmdIns As New MySqlCommand("INSERT INTO alarm (id, )
VALUES ('" & tagid & "'", cn)
            cmdIns.ExecuteNonQuery()
            If ConnectionState.Open Then
                cn.Close()
            End If

        Else
        End If
    End Sub

    Private Sub Label1_Click(ByVal sender As System.Object, ByVal e
As System.EventArgs) Handles Label1.Click

    End Sub

    Private Sub TextBox2_TextChanged(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles TextBox2.TextChanged

    End Sub
    Private Sub Label13_Click(ByVal sender As System.Object, ByVal e
As System.EventArgs) Handles Label13.Click

    End Sub

    Private Sub Label14_Click(ByVal sender As System.Object, ByVal e
As System.EventArgs) Handles Label14.Click

    End Sub

    Private Sub TextBox1_TextChanged(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles TextBox1.TextChanged

    End Sub
```

```
    Private Sub Label15_Click(ByVal sender As System.Object, ByVal e
As System.EventArgs) Handles Label15.Click
    End Sub
    Private Sub Label20_Click(ByVal sender As System.Object, ByVal e
As System.EventArgs) Handles Label20.Click
    End Sub
    Private Sub PictureBox1_Click(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles PictureBox1.Click
    End Sub
    Private Sub BtnAddOfficer_Click(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles BtnAddOfficer.Click
        add_newstaff.Show()
    End Sub
    Private Sub BtnAddInmate_Click(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles BtnAddInmate.Click
        add_newdoc.Show()
    End Sub

    Private Sub BtnOpList_Click(ByVal sender As System.Object, ByVal
e As System.EventArgs) Handles BtnOpList.Click
        Staff_List.Show()
    End Sub

    Private Sub BtnLogOut_Click(ByVal sender As System.Object, ByVal
e As System.EventArgs) Handles BtnLogOut.Click
        DT_Intro.Show()
        Me.Hide()
    End Sub

    Private Sub BtnInmLst_Click(ByVal sender As System.Object, ByVal
e As System.EventArgs) Handles BtnInmLst.Click
        Doc_List.Show()
    End Sub

    Private Sub TmrCall_Tick(ByVal sender As System.Object, ByVal e
As System.EventArgs) Handles TmrCall.Tick
        BtnStart_Click(sender, e)
    End Sub

    Private Sub ListView1_SelectedIndexChanged(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
ListView1.SelectedIndexChanged

    End Sub

    Private Sub Button3_Click_1(ByVal sender As System.Object, ByVal
e As System.EventArgs) Handles Button3.Click
        Search_Doc.Show()
    End Sub
    Private Sub TextBox5_TextChanged(ByVal sender As System.Object,
ByVal e As System.EventArgs)

    End Sub
End Class
```

Document List Window

```
Imports MySql.Data.MySqlClient
Imports System.Threading

Public Class Doc_List


    Dim strCn As String = "Database= dtraxx_db;Data
Source=localhost;User Id=root;Password="
    Private Strt As System.Threading.Thread
    Dim ID As ListViewItem


    Private Sub InmateList_Load(ByVal sender As System.Object, ByVal
e As System.EventArgs) Handles MyBase.Load

        InmList.Columns.Clear()
        InmList.Columns.Add("No", 30, HorizontalAlignment.Left)
        InmList.Columns.Add("Course Code", 90,
HorizontalAlignment.Left)
        InmList.Columns.Add("Course Name", 180,
HorizontalAlignment.Left)
        InmList.Columns.Add("Tag ID", 45, HorizontalAlignment.Left)
        InmList.Columns.Add("Location", 80, HorizontalAlignment.Left)
        InmList.Columns.Add("Lecturer's Name", 140,
HorizontalAlignment.Left)
        InmList.Columns.Add("Register Date", 120,
HorizontalAlignment.Left)



        Strt = New Thread(AddressOf Thread1)
        Strt.Start()
    End Sub

    Private Sub list()

        If Me.InvokeRequired Then
            Me.Invoke(New MethodInvoker(AddressOf list))
        Else
            'Establish connection
            Dim i As Integer = 0
            Dim No As Integer = 0

            Dim cn As New MySqlConnection(strCn)
            Dim cmd As New MySqlCommand("SELECT * FROM doc", cn)
            Dim myReader As MySqlDataReader


            InmList.Items.Clear()

            If cn.State = ConnectionState.Closed Then
                cn.Open()
            End If
```

```
            myReader = cmd.ExecuteReader()
            While myReader.Read

                No = No + 1
                ID = InmList.Items.Add(No.ToString)

ID.SubItems.Add(myReader.Item("Course_Code").ToString)


ID.SubItems.Add(myReader.Item("Course_Name").ToString)
                ID.SubItems.Add(myReader.Item("tag_id").ToString)
                ID.SubItems.Add(myReader.Item("location").ToString)
                ID.SubItems.Add(myReader.Item("lecturer").ToString)
                ID.SubItems.Add(myReader.Item("acqdate").ToString)
                'ID.SubItems.Add(myReader.Item("bed").ToString)


            End While

            cmd.Connection.Close()

            'coloring backgroung
            While i <= InmList.Items.Count - 1
                If i Mod 2 = 0 Then
                    InmList.Items(i).BackColor = Color.BlanchedAlmond
                Else
                    InmList.Items(i).BackColor = Color.White
                End If
                i = i + 1
            End While

        End If

    End Sub

    Sub Thread1()
        list()
    End Sub


    Private Sub InmList_SelectedIndexChanged(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
InmList.SelectedIndexChanged

    End Sub


End Class
```

Staff List Window

```
Imports MySql.Data.MySqlClient
Imports System.Threading

Public Class Staff_List

    Dim strCn As String = "Database=dtraxx_db;Data
Source=localhost;User Id=root;Password="
    Private Strt As System.Threading.Thread
    Dim ID As ListViewItem

    Private Sub OfficerList_Load(ByVal sender As System.Object, ByVal
e As System.EventArgs) Handles MyBase.Load
        OffList.Columns.Clear()
        OffList.Columns.Add("No", 30, HorizontalAlignment.Left)
        OffList.Columns.Add("Name", 180, HorizontalAlignment.Left)
        OffList.Columns.Add("Staff Number", 90,
HorizontalAlignment.Left)
        OffList.Columns.Add("Phone Number", 90,
HorizontalAlignment.Left)
        OffList.Columns.Add("Email Address", 160,
HorizontalAlignment.Left)

        Strt = New Thread(AddressOf Thread1)
        Strt.Start()
    End Sub

    Private Sub list()

        If Me.InvokeRequired Then
            Me.Invoke(New MethodInvoker(AddressOf list))
        Else
            'Establish connection
            Dim i As Integer = 0
            Dim No As Integer = 0

            Dim cn As New MySqlConnection(strCn)
            Dim cmd As New MySqlCommand("SELECT * FROM staff", cn)
            Dim myReader As MySqlDataReader


            OffList.Items.Clear()

            If cn.State = ConnectionState.Closed Then
                cn.Open()
            End If
```

```
            myReader = cmd.ExecuteReader()
            While myReader.Read

                No = No + 1
                ID = OffList.Items.Add(No.ToString)
                ID.SubItems.Add(myReader.Item("Name").ToString)
                ID.SubItems.Add(myReader.Item("staff_id").ToString)


ID.SubItems.Add(myReader.Item("Phone_Number").ToString)
                ID.SubItems.Add(myReader.Item("email").ToString)


            End While


            cmd.Connection.Close()

            'coloring backgroung
            While i <= OffList.Items.Count - 1
                If i Mod 2 = 0 Then
                    OffList.Items(i).BackColor = Color.BlanchedAlmond
                Else
                    OffList.Items(i).BackColor = Color.White
                End If
                i = i + 1
            End While

        End If

    End Sub

    Sub Thread1()
        list()

    End Sub


    Private Sub OffList_SelectedIndexChanged(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
OffList.SelectedIndexChanged


    End Sub

End Class
```

Add New Staff Window

```
Imports System
Imports System.ComponentModel
Imports System.Threading
Imports System.Windows.Forms
Imports System.Runtime.InteropServices
Imports Microsoft.VisualBasic
Imports System.IO
Imports MySql.Data.MySqlClient

Public Class add_newstaff

    Dim strCn As String = "Database= dtraxx_db;Data
Source=localhost;User Id=root;Password="

    Private Sub addofficer_Load(ByVal sender As System.Object, ByVal
e As System.EventArgs) Handles MyBase.Load


    End Sub


    Private Sub btn_confirm_Click(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles btn_confirm.Click
        Dim rowsCheck As Integer = 0
        Dim id As String = TextBox1.Text.ToString
        Dim name As String = TextBox2.Text.ToString
        Dim pswd As String = TextBox3.Text.ToString
        Dim stid As Integer = TextBox7.Text.ToString
        Dim sid As String = TextBox4.Text.ToString
        Dim hp As String = TextBox5.Text.ToString
        Dim email As String = TextBox6.Text.ToString

        Dim cn As New MySqlConnection(strCn)

        Dim cmdCheck As New MySqlCommand("SELECT * FROM staff WHERE
id = '" & id & "'", cn)
        Dim daCheck As New MySqlDataAdapter
        Dim dsCheck As New DataSet
        Dim dtCheck As New DataTable


        cn.Open()

        'Check from table (search)
        Try
            With daCheck
                .SelectCommand = cmdCheck
                .Fill(dsCheck, "searchresult")
            End With
            dtCheck = dsCheck.Tables("searchresult")
            rowsCheck = dtCheck.Rows.Count()

            'If unable to connect, show error!
```

```
        Catch ex As Exception
            MsgBox("Error: " & ex.Source & ": " & ex.Message,
MsgBoxStyle.OkOnly, "Connection Error !!")
        End Try

        'If already connected, close connection
        If ConnectionState.Open Then
            cn.Close()
        End If


        If rowsCheck = 0 Then
            'Open connection
            cn.Open()

            Dim cmdIns As New MySqlCommand("INSERT INTO staff (id,
name, Password, staff_tid, staff_id, phone_number, email) VALUES ('"
& id & "','" & name & "','" & pswd & "', '" & stid & "', '" & sid &
"', '" & hp & "', '" & email & "')", cn)
            cmdIns.ExecuteNonQuery()
            If ConnectionState.Open Then
                cn.Close()
            End If

                        MsgBox(id + "/" + name + " has been
registered successfully")

        Else
            'If already existing officer, show error
            MsgBox("Error!!! " + id + "is already registered in the
database")

        End If

    End Sub


    Private Sub btn_Oclose_Click(ByVal sender As System.Object, ByVal
e As System.EventArgs) Handles btn_Oclose.Click
        TextBox1.Text = ("")
        TextBox2.Text = ("")
        TextBox3.Text = ("")
        TextBox7.Text = ("")
        TextBox4.Text = ("")
        TextBox5.Text = ("")
        TextBox6.Text = ("")

    End Sub

    Private Sub TextBox1_TextChanged(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles TextBox1.TextChanged

    End Sub
End Class
```

Add New Document Window

```
Imports MySql.Data.MySqlClient
Public Class add_newdoc

    Private Sub Submit_Click(ByVal sender As System.Object, ByVal e
As System.EventArgs) Handles Submit.Click
        Dim ccode As String = TextBox2.Text.ToString
        Dim rowsCheck As Integer = 0
        Dim cname As String = TextBox1.Text.ToString
        Dim locate As String = TextBox5.Text.ToString
        Dim tagid As Integer = TextBox4.Text.ToString
        Dim lecturer As String = TextBox6.Text.ToString
        Dim acqdate As String = txtacq.Text.ToString

        Dim strCn As String = "Database= dtraxx_db;Data
Source=localhost;User Id=root;Password="
        Dim cn As New MySqlConnection(strCn)
        Dim cmdCheck As New MySqlCommand("SELECT * FROM doc WHERE
Course_Code = '" & ccode & "'", cn)
        Dim daCheck As New MySqlDataAdapter
        Dim dsCheck As New DataSet
        Dim dtCheck As New DataTable
        cn.Open()
        Try
            With daCheck
                .SelectCommand = cmdCheck
                .Fill(dsCheck, "searchresult")
            End With
            dtCheck = dsCheck.Tables("searchresult")
            rowsCheck = dtCheck.Rows.Count()
        Catch ex As Exception
            MsgBox("Error: " & ex.Source & ": " & ex.Message,
MsgBoxStyle.OkOnly, "Connection Error !!")
        End Try
        If ConnectionState.Open Then
            cn.Close()
        End If


        If rowsCheck = 0 Then

            cn.Open()
            Dim cmdIns As New MySqlCommand("INSERT INTO doc
(Course_Code, Course_Name, Location,  tag_id, lecturer, acqdate)
VALUES ('" & ccode & "','" & cname & "','" & locate & "','" & tagid &
"','" & lecturer & "','" & acqdate & "')", cn)
            cmdIns.ExecuteNonQuery()

            If ConnectionState.Open Then
                cn.Close()
            End If
            MsgBox(ccode + "/" + cname + " has been registered
successfully")
```

```
        Else
            MsgBox("Error!!! " + ccode + "is already registered in
the database")

        End If
    End Sub

    Private Sub Button1_Click(ByVal sender As System.Object, ByVal e
As System.EventArgs)
        Me.Close()
    End Sub

    Private Sub addinm_Load(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles MyBase.Load
    End Sub

    Private Sub Label3_Click(ByVal sender As System.Object, ByVal e
As System.EventArgs)

    End Sub

    Private Sub TextBox4_TextChanged(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles TextBox4.TextChanged

    End Sub

    Private Sub btn_Clear1_Click(ByVal sender As System.Object, ByVal
e As System.EventArgs) Handles btn_Clear1.Click
        TextBox2.Text = ("")
        TextBox1.Text = ("")
        TextBox5.Text = ("")
        TextBox4.Text = ("")
        TextBox6.Text = ("")
        txtacq.Text = ("")

    End Sub

    Private Sub TextBox2_TextChanged(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles TextBox2.TextChanged
    End Sub

    Private Sub TextBox1_TextChanged(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles TextBox1.TextChanged
    End Sub

    Private Sub Label5_Click(ByVal sender As System.Object, ByVal e
As System.EventArgs) Handles Label5.Click

    End Sub
    Private Sub Label2_Click(ByVal sender As System.Object, ByVal e
As System.EventArgs) Handles Label2.Click

    End Sub
End Class
```

Search Document Window

```
Imports System
Imports System.ComponentModel
Imports System.Threading
Imports System.Windows.Forms
Imports System.Runtime.InteropServices
Imports Microsoft.VisualBasic
Imports AW_API_NET
Imports System.IO
Imports MySql.Data.MySqlClient

Public Class Search_Doc

    Dim Hconn As IntPtr
    Dim readerIP(20) As Byte
    Dim readerPort As UInt16
    Dim commPort As UInt32
    Dim commBaud As UInt32
    Dim myPKTID As Integer
    Dim registered As Boolean
    Dim strhttp As String

    Dim ActiveWaveAPI As AW_API_NET.APINetClass = New
AW_API_NET.APINetClass
    Dim ReaderEventHandler As AW_API_NET.fReaderEvent
    Dim TagEventHandler As AW_API_NET.fTagEvent
    Dim ipIdx As Integer = 0
    Dim pubReaderID As Integer
    Dim c As Integer
    Dim txtSender As String

    Dim tagID As ListViewItem
    Private Strt As System.Threading.Thread
    Dim strCn As String = "Database=dtraxx_db;Data
Source=localhost;User Id=root;Password="
    Dim selID As String

    Private Sub Button1_Click(ByVal sender As System.Object, ByVal e
As System.EventArgs) Handles Button1.Click

        Dim courseCode As String = ""
        Dim courseName As String = ""
        Dim tagID As String = ""
        Dim location As String = ""
        Dim enrDate As String = ""
        If Me.TextBox1.Text <> "" Then
        courseCode = Me.TextBox1.Text
```

```
            Dim cn As New MySqlConnection(strCn)
            Dim cmd As New MySqlCommand("SELECT Course_Name, tag_id,
 location, acqdate FROM doc WHERE Course_Code = '" & courseCode & "'",
 cn)
            Dim rdr As MySqlDataReader

            cn.Open()
            rdr = cmd.ExecuteReader

            While rdr.Read
                courseName = rdr.Item("Course_Name").ToString
                tagID = rdr.Item("tag_id").ToString
                location = rdr.Item("location").ToString
                enrDate = Format(rdr.Item("acqdate"), "yyyy-MM-dd
 h:mm:ss")
            End While

            Doc_Details.txtCcode.Text = courseCode
            Doc_Details.txtCname.Text = courseName
            Doc_Details.txtTagID.Text = tagID
            Doc_Details.txtLoc.Text = location
            Doc_Details.txtDateTime.Text = enrDate

            'doc_details.Show()
            Doc_Details.Show()

  Else

            MsgBox("Please enter the course code!",
MsgBoxStyle.Critical, "D'TraXX")
        End If

    End Sub


    Private Sub Button2_Click(ByVal sender As System.Object, ByVal e
As System.EventArgs) Handles Button2.Click
        Me.Close()
    End Sub

    Private Sub Search_Doc_Load(ByVal sender As System.Object, ByVal
e As System.EventArgs) Handles MyBase.Load

    End Sub

End Class
```

Document Details Window

```
Imports MySql.Data

Public Class Doc_Details

    Private Sub documentDetails_Load(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles MyBase.Load
        Dim SqlConnStr As String = "Database=dtraxx_db;Data
Source=127.0.0.1;User Id=root;Password="
        Dim SqlConn As New MySqlClient.MySqlConnection(SqlConnStr)
        Dim SqlAdapter As New MySqlClient.MySqlDataAdapter
        Dim SqlDataset As New DataSet
        SqlAdapter.SelectCommand = New
MySqlClient.MySqlCommand("SELECT * FROM `doc` WHERE `Course_Code` =
'" & Me.Tag & "' LIMIT 0,1", SqlConn)
        SqlAdapter.SelectCommand.Connection.Open()
        SqlAdapter.Fill(SqlDataset)
        SqlAdapter.SelectCommand.Connection.Close()

        Dim UserQueryResult As DataTable = SqlDataset.Tables(0)
    End Sub

    Private Sub btnClose_Click(ByVal sender As System.Object, ByVal e
As System.EventArgs)
        Me.Close()
    End Sub

    Private Sub btn_close_Click(ByVal sender As System.Object, ByVal
e As System.EventArgs) Handles btn_close.Click
        Me.Close()
    End Sub

    Private Sub txtDateTime_TextChanged(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
txtDateTime.TextChanged

    End Sub

    Private Sub Doc_Details_Load(ByVal sender As System.Object, ByVal
e As System.EventArgs) Handles MyBase.Load

    End Sub

    Private Sub txtCcode_TextChanged(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles txtCcode.TextChanged

    End Sub
End Class
```